Reentry Vehicle Flight Controls Design Guidelines: Dynamic Inversion

Daigoro Ito
Jennifer Georgie
John Valasek
Donald T. Ward
Flight Simulation Laboratory
Texas Engineering Experiment Station
Texas A&M University

March 2002

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

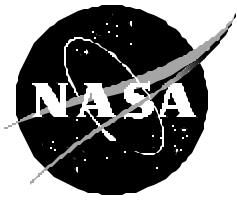- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at *http://www.sti.nasa.gov*

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Telephone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace
      Information
  7121 Standard Drive
  Hanover, MD 21076-1320

Reentry Vehicle Flight Controls Design Guidelines: Dynamic Inversion

Daigoro Ito
Jennifer Georgie
John Valasek
Donald T. Ward
Flight Simulation Laboratory
Texas Engineering Experiment Station
Texas A&M University

Contents

Tables

Figures

Nomenclature and Acronyms

Symbols	Definition
b	wingspan
C_l	nondimensional rolling coefficient
C_m	nondimensional pitching coefficient
C_n	nondimensional yawing coefficient
CV	control variable
g	gravitational acceleration
h	altitude
I	moment of inertia
K	gain
L_p	roll rate stability derivative
L_r	roll rate derivative with respect to yaw rate
L_β	roll rate derivative with respect to sideslip angle
$L_{\delta a}$	roll rate derivative with respect to aileron deflection angle
$L_{\delta r}$	roll rate derivative with respect to rudder deflection angle
LCV	roll control variable
M_∞	free stream Mach number
M_α	pitch rate derivative with respect to angle-of-attack
M_q	pitch rate stability derivative
$M_{\delta e}$?	pitch rate control derivative
MCV	pitch control variable
N_p	yaw rate stability derivative
N_r	yaw rate derivative with respect to yaw rate
N_β	yaw rate derivative with respect to sideslip angle
$N_{\delta a}$	yaw rate derivative with respect to aileron deflection angle
$N_{\delta r}$	yaw rate derivative with respect to rudder deflection angle
NCV	yaw control variable
p	body-axis roll rate
p_s	stability-axis roll rate
q	body-axis pitch rate
$\bar{q}$	dynamic pressure
r	body-axis yaw rate
r_s	stability-axis yaw rate
S	wing reference area
U	control vector
V	velocity
V_{CO}	airspeed at which pitch rate and normal acceleration (at constant α) make equal contributions to the controlled variable
X	state vector
Y	output vector

Greek Symbols	Definition
α	angle-of-attack
β	sideslip angle
χ	heading angle

δ_e	elevator deflection angle
δ_a	aileron deflection angle
δ_r	rudder deflection angle
ϕ	bank angle
γ	flight path angle
η	east position
λ_{n_z}	n_z bandwidth
μ	structured singular value
θ	attitude angle
σ	singular value
ω	frequency
ω_n	natural frequency
ξ	north position
ζ	damping ratio
$|\cdot|$	absolute value

Acronyms	Definition
ARE	algebraic Ricatti equation
CAP	control anticipation parameter
CRV	crew return vehicle
CV	control variable
DI	dynamic inversion
FML	Flight Mechanics Laboratory
HARV	high angle-of-attack research vehicle
JSC	Johnson Space Center
LFT	linear fractional transformation
LMI	linear matrix inequality
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
MACH	Multi-application Control
MIMO	multiple input, multiple output
PC	personal computer
PI	proportional integral
RMS	root mean square
SES	shuttle engineering simulator
SGI	Silicon Graphics Incorporated
SISO	single input, single output
TAMU	Texas A&M University

Subscripts and Superscripts	Definition
^	estimated value
–	upper bound
.	time derivative
0	nominal value
Aug	augmented system
c	compensator
cmd	commanded value
des	desired value

dist	disturbance
IL	inner loop
max	maximum value
meas	measured value
$\bar{q}$	dynamic pressure
s	measurement
T	total
v	measurement noise
w	process noise

1   Introduction to the Problem

1.1   Purpose of the Document

This document is a product of a research project initiated in February 1999 by the X-38 Flight Controls Branch at the NASA Johnson Space Center (JSC). Funded by NASA Grant NAG9-1085, the effort was associated with the Flight Mechanics Laboratory (FML) of the Texas Engineering Experiment Station – the research arm of the Dwight Look College of Engineering at Texas A&M University (TAMU). One of the tasks of the unsolicited proposal that led to this grant was to provide a set of design guidelines that could be used in future by JSC. The subject of these guidelines was to be a flight control design for vehicles operating across a broad flight regime and with highly nonlinear physical descriptions of motion. The guidelines specifically were to address the need for reentry vehicles that could operate, as the X-38 does, through reentry from space to controlled touchdown on the Earth's surface. The latter part of controlled descent was to be achieved by parachute or paraglider – or by an automatic or a human-controlled landing similar to that of the space shuttle Orbiter.

Since these guidelines address the specific needs of human-carrying (but not necessarily piloted) reentry vehicles, they deal with highly nonlinear equations of motion, and their generated control systems must be robust across a very wide range of physics. Thus, this first-generation document deals almost exclusively with some form of dynamic inversion (DI), a technique that has been widely studied and applied within the past 25 to 30 years. Comprehensive and rigorous proofs now exist for transforming a nonlinear system into an equivalent linear system. (Called either feedback linearization or DI, it is based on the early papers of Krener and Brockett[1,2].) At about the same time, theoretical advances essentially completed the background for ensuring the feedback control laws that make prescribed outputs independent of important classes of inputs; namely, disturbances and decoupled control effectors. These two vital aspects of control theory – noninteracting control laws and the transformation of nonlinear systems into equivalent linear systems – are embodied in what is often called DI. Falb and Wolovich[3] considered noninteractions as a facet of linear systems theory. Singh, Rugh, Freund, and Porter[4,5,6] extended these notions into nonlinear systems. Isidori and his colleagues[7,8] contributed significantly to DI theory by using mathematical notions from differential geometry. Balas and his colleagues applied these ideas to a variety of aerospace flight control system designs – including the F-18 high angle-of-attack research vehicle (HARV)[9] as well as to the X-38[10] itself. They also provided powerful, commercially available software tools[11] that are widely used by control design practitioners. Though there is no doubt that the mathematical tools and underlying theory are available to industry and government agencies, there are open issues as to the practicality of using DI as the only (or even the primary) design approach for reentry vehicles. Our purpose, therefore, is to provide a set of guidelines that can be used to determine the practical usefulness of the technique.

This document will answer the following questions related to four main topics:

1. If we use DI as our primary design method, what tools are available to implement the design tasks?

2. How easy is it to obtain and to learn to use these tools? Can an entry-level (an undergraduate) engineer be expected to be familiar enough with the tools to be productive without receiving specialized training and consulting help?

3. Is it easy to convey the value of using DI? How does a design group communicate the validation of systems modeled with this modern control technique?
4. What form of robustness analysis is appropriate? Is more than one technique worth considering?

Section 2 of this report addresses the first question by first summarizing the value of three tools used by TAMU FML engineers – MACH [Mutli-Application Control], MATLAB, and batch simulations. This section goes on to investigate and explore the available forms of robustness analysis (question 4) as the forms relate to practical uncertainties and disturbances. Section 2 concludes with first thoughts on how we would go about evaluating the various tools.

Section 3 addresses how DI is achieved from the perspective of new graduate students who has to teach themselves these techniques. It is hoped that later studies will expand and extend this validation process to show that less-sophisticated talent can also successfully complete workable designs.

Section 4 illustrates the simulation component buildup surrounding DI, and it applies DI to the X-38 reentry vehicle model in three separate examples. The first tests a DI controller against a nonlinear MATLAB simulation to evaluate performance; the second and third present longitudinal and lateral/directional DI controller designs, respectively.

Section 5 describes two different controller analysis techniques and analyzes DI controllers using both methods. The controller analysis techniques addressed in this section include μ-analysis and linear quadratic performance index analysis.

Section 6 provides a summary of the theoretical background needed to understand some of the DI design procedures and to complete at least elementary robustness analyses of the DI system.

Finally, Section 7 is a fairly extensive list of references used to prepare this report. Although the bibliography is not comprehensive, it does include much of the classical work that has been done to this point.

2 Synthesis Procedure

Synthesis is the process by which the components or elements of a system are brought together by a designer to accomplish the tasks under consideration. The trick is to be sure that the individual parts are integrated in such a way that the sum of the parts produces an outcome greater than the individual contributions of the parts. This "synergy" is a result of an integrated design. Integration begins with the process used, and depends strongly on the tools available.

In this first iteration of our design guidelines, we will consider three sets of software tools; i.e., MACH, MATLAB, and batch simulations. MATLAB is a widely used commercial software package for control system design that has both a command line and a graphical user interface. It is also relatively easy to use, and many colleges and universities teach undergraduate courses that integrate MATLAB-based problems into their pedagogy. Moreover, MATLAB has a number of specialized toolkits that directly address matrix algebra and modern control system design, including DI and techniques often used to analyze the robustness of such designs. This set of tools is quite extensively documented; indeed, MATLAB has steadily evolved and been improved over several years of commercial usage.

MACH is a set of proprietary software tools developed and used (but not sold commercially) by Honeywell that directly address some of issues common to DI. One of the key questions we want to answer in this report is: Is it feasible for a relative beginner to build up DI models without using tools such as MACH? Or, is MACH indispensable to the efficient generation of DI modules?

Finally batch simulation, which can be done (at least partially) within MATLAB's Simulink module, is a software tool that requires some attention. It is doubtful that a control system designer today would attempt to produce a flight-worthy system without first generating at least a mathematical model of the specific system under consideration. We used the shuttle engineering simulator (SES) as the basis for our batch simulation buildup. As is almost always the case, keeping the simulation current as the vehicle (in this case, the X-38) design evolves is a recurring headache. As new data become available, the simulation has to be updated. ***Lesson Learned 1****: Set up a procedure early in the process for updating and formatting aerodynamic (and other) databases.* A corollary to this is that time and resources must be devoted to maintaining these databases or all facets of the program will suffer. Flight control design cannot proceed efficiently without this effort.

2.1 Tools

2.1.1 *MATLAB*

As mentioned earlier, MATLAB is one of the most widely used commercial software packages available for control system design. So, it and its companion product, Simulink, are used extensively in this study. The DI controllers are developed in MATLAB, and simulations are run within the Simulink environment. This simulation development process and its results are at the core of this study. Detailed linear robustness analyses of the example controller for the X-38 are also made easier by MATLAB. Two different MATLAB toolboxes will facilitate μ-analysis: the (1) Robust Control Toolbox and (2) μ-Synthesis and Analysis Toolbox. Obviously, the latter is intended for μ-analysis since it is dedicated to that process. Hence, a number of useful functions are readily available and packaged with a detailed instruction manual. A detailed discussion of the underlying theory of μ-analysis is given in Section 5.1.

2.1.2 *Multi-Application Control*

One of the spacecraft controllers using the DI approach is based on MACH. MACH is a proprietary software package developed by Honeywell that was previously applied to several flight control designs such as the F-18 HARV and the X-29 aircraft. Its basic structure consists of an inner-

loop DI controller wrapped around an outer-loop classical proportional integral (PI) controller. Figure 2.1 below shows the similarity between the MACH system structure and the controller used in the first example (see Section 4.6). Although slight differences exist, these are associated primarily with different definitions for the control variables (CVs).
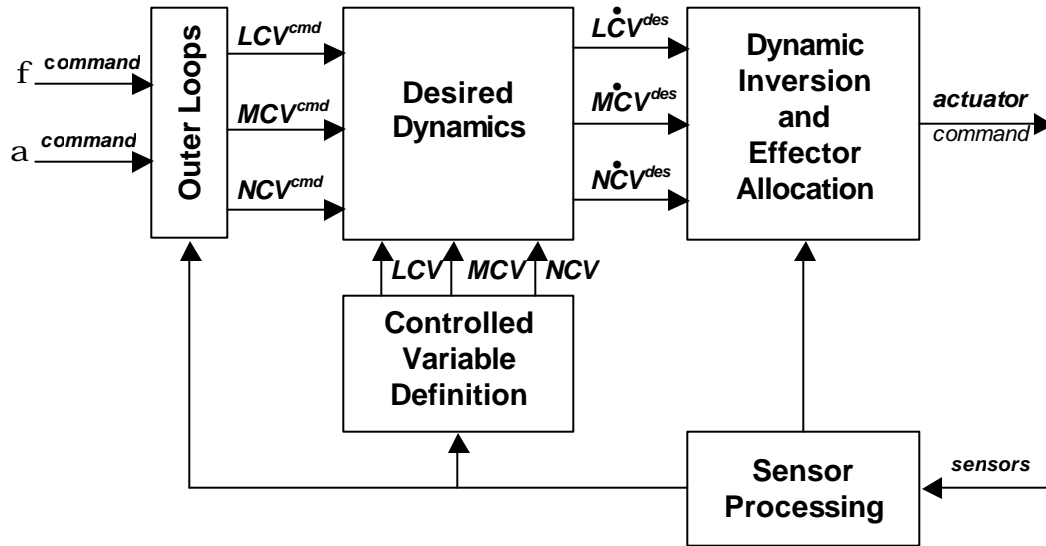
Figure 2.1 Typical MACH system structure

Since MACH is proprietary to Honeywell, implementation details cannot be presented in a document of unlimited distribution such as these guidelines. Details are sketchy in any event; and other than a comprehensive outline of the MACH structure, the code is not used broadly as a design tool in this version of our guidelines. Later implementations may include more on the MACH software if it becomes obvious that the code is useful to the overall design process. For purposes of this document, we will therefore focus on demonstrating that DI can be successively implemented with other tools and procedures with only a moderately intense learning effort on the part of the analyst. A brief comparison of a MACH controller and a Simulink example are presented in Section 4.4. This comparison primarily highlights the differences.

2.1.3   Batch Simulation

The batch mode of SES X-38-V201 version 1.3 is used in this study. This version contains the shuttle-derived classical controller designed at JSC by John Ruppert. Although different versions of MACH have already been implemented for the X-38-V132; this is the only version already implemented that uses the V201 database prior to release of the MACH controller in late 1999. This SES version was thus our only available choice since the scope of our study was to examine the characteristics of a DI controller throughout the entire X-38 flight envelope (i.e., from hypersonic through subsonic flight regimes of anticipated trajectories).

By using batch implementation, a nominal case is executed to obtain essential vehicle properties (aerodynamic coefficients, moment/product of inertia, etc.). The data thus obtained are then incorporated into the Simulink-based DI controllers. A few attempts are also included in which we began examining uncertainty in the mass properties of the X-38 by varying these parameters slightly

(Section 4.6). However, time constraints as well as the complexity of the SES batch simulation limited the number we performed of these runs.

2.2   Specifications

The controller design procedure is normally iterative and centers around designing a controller that satisfies a set of design specifications. These specifications can be provided in the time domain, the frequency domain, or both. It is important to note that the type of input (i.e., impulse, step, ramp, sinusoid) must be specified. Examples of these are introduced and discussed below.

2.2.1   Time Domain

Time domain inputs, such as a step input, can be used to evaluate system characteristics such as damping, natural frequency, overshoot, etc. Initial condition or impulse excitations are particularly useful in evaluating the damping of rate variables. Such time domain controller responses can be evaluated through simulation in the Simulink environment, for example.

2.2.2   Frequency Domain

The response of a linear system to a sinusoidal input is referred to as the system's frequency response. Frequency domain specifications are concerned with the response of a system to frequency varying inputs, most often of the sinusoidal type. Typical specifications are gain margin, phase margin, and bandwidth. Gain margin is the amount by which system gain can increase before the system becomes neutrally stable. Phase margin is the amount by which phase lag can increase before the system becomes neutrally stable. Bandwidth – defined as the maximum frequency at which system output will satisfactorily track a sinusoid input – is basically a frequency domain measure of response speed. It is therefore akin to the time-domain specification of rise time. Frequency domain specifications are important to multiple-input, multiple-output (MIMO) robust controller design since most available methods are based in the frequency domain and thus use some or all of the frequency domain specifications.

2.3   Uncertainty Modeling

Actual controllers are expected to perform well for an entire class of transfer functions representing the range of plant dynamics and operating environment. Since it is impossible to analytically or empirically model with 100% accuracy a dynamic system and the effects of its operating environment, uncertainty modeling plays an important role in controller design and analysis. But even when applying optimal control techniques, the resulting controller designs will not be truly "optimal" because

- operating environments can introduce undesirable/unknown performance.
- the system is inherently nonlinear (*EXAMPLE*: Coulomb friction, hysteresis, backlash, and deadbands).
- physical components are subject to wear and failure.
- there are limitations to implementation (*EXAMPLE*: computational delays).

Uncertainties are broadly classified in two categories – structured and unstructured – both of which are usually present in any given physical system. The key to successful uncertainty modeling (and, thus, to robust controller design) is to recognize to which category a particular type of uncertainty belongs and then to determine the characteristics of that uncertainty. Further, the majority of robust control

techniques require uncertainty be modeled entirely in the frequency domain. These topics are outlined in the following sections.

2.3.1 Structured

Structured uncertainties can be modeled and have relatively well-known bounds and ranges. Parametric uncertainties arise from and include

- control effectiveness,
- aerodynamics,
- mass, and
- inertia.

Prediction Accuracy of Aircraft Stability Derivatives

$$Lift\ Curve\ Slope\ C_{L_a}\ \pm\ 5\%$$

$$Pitch\ Damping\ C_{m_q}\ \pm\ 20\%$$

$$Yawing\ Moment\ Due\ To\ Roll\ Rate\ C_{n_p}\ \pm\ 90\%$$

Prediction Accuracy of Spacecraft Inertias

$$I_{nn}\ \pm\ 10\%$$

$$I_{np}\ \pm\ 10\%$$

Parametric uncertainties are important; but since by definition they can be understood and modeled, they can usually be analyzed and handled. Nonparametric uncertainties are potentially more dangerous because they are not as well understood and are difficult or impossible to model accurately. However, those that can be represented as some type of stochastic process can be easily incorporated into the design model. The key issue is to determine the relative magnitude of nonparametric uncertainties. In general, small nonparametric uncertainties cause small errors while large nonparametric uncertainties cause large errors. It is also important to determine how quickly nonparametric uncertainties vary.

2.3.2 Unstructured

Unstructured uncertainties are those for which generally little to no knowledge is possessed. They are usually not modeling-related nor can they be modeled at all. Nonparametric unstructured uncertainties include

- high-frequency unmodeled dynamics,
- actuator dynamics,
- structural vibrations,
- measurement noise,
- round-off error and truncation, and
- sampling delay.

Since retention of full, nonlinear dynamics severely restricts the number of synthesis techniques presently available, linearization of actually nonlinear dynamics is often required. So, approximations are inherent and introduce uncertainty. Actuators also fall into this category for the same reason. Uncertainty due to structural vibrations and measurement noise can be represented with a certain degree of accuracy when experimental data is available. In the absence of experimental data or when a simpler representation is wanted, measurement noise is often approximated as a sine wave. Round-off error and truncation are extremely difficult means of representing uncertainties. No widely accepted standard method exists for them.

2.3.3   Frequency Domain

Classical Control addresses the issue of uncertainty by assuming that all types of uncertainties in the system cause only gain changes, or phase changes, to occur. Robust Modern Control takes a frequency domain approach using transfer functions in the S-domain such that certain types of modeling errors are assumed to have certain frequency effects. Since parametric modeling errors are structured uncertainties with known bounds, they are assumed to cause low-frequency effects. Consequently, neglected and possibly higher-order dynamics are assumed to cause high-frequency effects. Unstructured uncertainties, which are not well understood, represent systems in the frequency domain whose frequencies simply are assumed to lie between some upper and lower bound. Additive uncertainty is used to model errors in neglected high-frequency dynamics; this represents the *absolute* error in the model. Multiplicative uncertainty, which is used to model errors in actuators or sensor dynamics, represents the *relative* error in a model. This latter type of uncertainty is most useful in robustness analysis and design.

2.4   Disturbances

Disturbance rejection properties to exogenous disturbances – e.g., gusts, turbulence, wind shear – are particularly critical in flight control system design. By definition, an exogenous input is one that a controller cannot manipulate. These unstructured uncertainties are stochastic processes and, as such, are best represented as stochastic models in terms of mean and variance. The standard gust and turbulence models, due to Von Karman and Dryden, are empirically based and directly applicable to both controller design and controller analysis.

2.5   Dynamic Inversion Synthesis

DI synthesis is a controller synthesis technique by which existing deficient, or undesirable, dynamics are canceled out and replaced by desirable dynamics. Cancellation and replacement are achieved through careful algebraic selection of the feedback function. For this reason, this methodology is also called *feedback linearization*. It applies to both single-input, single-output (SISO) and MIMO systems, provided the control effectiveness function (in the SISO case) or the control influence matrix (in the MIMO case) is invertible. The method works for both full-state feedback (input-state feedback linearization) and output feedback (input-output feedback linearization). A fundamental assumption in this methodology is that plant dynamics are perfectly modeled and can be canceled exactly. In practice this assumption is not realistic, so the new dynamics require some form of robust controller (see Section 2.6.1) to suppress undesired behavior due to plant uncertainties. Examples of DI synthesis are shown in Chapter 3.

2.6   Robustness

Compensators are designed to satisfy specified requirements for steady-state error, transient response, stability margins, or closed-loop pole locations. Meeting all objectives is usually difficult because of the various tradeoffs that have to be made and because of the limitations of design techniques. For example, although classical root locus design places a pair of complex conjugate poles to meet transient response specifications, the designer has little control over the location of all other poles and zeros. The particular property that a control system must have to operate properly in realistic situations is called *robustness*. A control system that possesses both good *disturbance rejection* and *low sensitivity* is said to be *robust*. Disturbance rejection is the ability to maintain good regulation (tracking) in the presence of disturbance signals. Low sensitivity is the ability to maintain good

regulation (tracking) in the presence of changes in plant parameters. Mathematically, this means that a controller must operate satisfactorily for not just one plant but for a family or a set of plants.

Robustness is divided into two distinct yet related categories: *stability robustness* and *performance robustness*. Stability robustness is the ability to guarantee closed-loop stability in spite of parameter variations and high-frequency unmodeled dynamics. It is important to note that relative stability, not absolute stability, is of interest in this context. Performance robustness is the ability to guarantee acceptable performance (settling time, overshoot, etc.) even although the system may be subject to disturbances. The Classical Control method quantifies robustness through gain margin and phase margin. Modern Control techniques use the structured singular value analysis of Section 6.2.3 to quantify robustness. In the MIMO case, both the maximum and the minimum singular values are measures of the amplification and attenuation, respectively, of the transfer function matrices that represent the family or set of plants of a system. Section 5.1 presents this robustness technique and demonstrates how to perform the analysis and interpret the results.

2.6.1 μ-Synthesis and H_∞

Structured singular value synthesis, or μ-synthesis, is a multivariable design method that can be used to directly optimize robust performance. It involves both μ-analysis and H_∞ synthesis. Performance specifications are weighted transfer functions describing the magnitude and frequency content of control inputs, exogenous inputs, sensor noise, tracking errors, actuator activity, and flying qualities. A family of models (consisting of a nominal model plus structured perturbation models) is used with magnitude bounds and frequency content specified using weighted transfer functions. All of this is wrapped into a single standard interconnection structure that is then operated upon by the algorithm.

The H_∞ control controller design methodology is a frequency domain optimization for robust control systems. H_∞ is defined as the space of proper and stable transfer functions – i.e., transfer functions with a number of zeros less than or equal to the number of poles. The objective is to minimize the H_∞ norm. Physically, this corresponds to minimizing the peak value in the Bode magnitude plot of the transfer function in the SISO case or the singular value plot in the MIMO case. There are certain advantages in minimizing the infinity-norm. These are

- The infinity-norm is the energy gain of the system. By comparison, the Linear Quadratic Gaussian (LQG) technique minimizes the 2-norm, which is not a gain.
- The infinity-norm minimizes the worst-case root mean square (RMS) value of the regulated variables when the disturbances have unknown spectra. The 2-norm minimizes the RMS values of the regulated variables when the disturbances are unit-intensity, white noise processes.
- H_∞ control results is guaranteed stability margins (and is therefore robust), whereas LQG has no guaranteed margins.

As in the Linear Quadratic Regulator (LQR)/LQG methodology, H_∞ is iterative. In the standard problem, the solution for the infinity-norm is iterated upon until it is less than a specified scalar value, gamma – known as the gamma iteration. In the optimal problem, the infinity-norm is progressively reduced until a solution does not exist. In the H_∞ control problem, the weights are the only design parameters the user must specify. Constant weights are used for scaling inputs and outputs. Transfer function weights are used to shape the various measures of performance in the frequency domain; weights are also used to satisfy the rank conditions. Proper selection of weights depends a great deal on understanding both the modeling process and the physics of the problem.

Necessary conditions for a solution are the ability to stabilize and detect the system; to perform various rank requirements on system matrices; and to ensure that the transfer function between exogenous system inputs and the outputs remains nonzero at high frequencies. This last condition, which is often violated, occurs because the transfer function is strictly proper; i.e., has more poles than zeros.

Solutions to H_∞ and LQG problems are very similar. Both use a state estimator and feed back the estimated states, and both solve two Ricatti equations to compute controller and estimator gains. The difference in the solutions lies in the coefficients of the Ricatti equation and in an extra term in the H_∞ solution. Examples of this methodology are presented in Chapter 5.

2.7   Validation

Validation – which consists of an attempt to match outputs between two different control and simulation software packages for the same control inputs, and for the same controller structure and gains – was performed on all examples in this document to ensure as much fidelity as reasonably possible. The degree of fidelity depends on the purpose of the example, the software tool used to synthesize and simulate the example, the operating system and language, and the platform on which the example was being run.

2.7.1   MATLAB versus MACH

MATLAB and MACH have similar structures that, in theory, should permit good validation. MATLAB was run on a personal computer (PC) and MACH was run on a UNIX-based workstation. The difficulty involved with this validation effort stemmed from a lack of understanding of the MACH code itself due to a lack of documentation. Although agreement between the two codes was generally good, it was inadequate for in-depth investigations and research.

2.7.2   MATLAB versus Batch Simulation

MATLAB was run on a PC, and the SES batch simulation was run on a Silicon Graphics Incorporated (SGI) UNIX workstation. Because of adequate documentation and open access to the SES source code, validation between these two codes proceeded rapidly and with excellent agreement. These two software codes form the basis for all of the controller design research presented in this document.

3   Applying Dynamic Inversion

3.1   Introduction and Philosophical Approach

This section shows how DI is applied to a relatively simple aircraft control problem. As will be explained in more detail in Section 4, since the concept of DI itself is quite simple, a controller can be designed in many different ways. For example, the controller might be either linear or nonlinear. Also, a DI controller is not limited to a first-order inversion. It can take on higher-order forms as well. This chapter describes one way of designing a DI-based controller. The steps taken in completing this design are carefully delineated in the hope that a step-by-step outline will help others design DI-based controllers.

First, a brief outline of the DI process will be given to quickly review the concept, followed by a detailed description of how to design each controller component. Then, aircraft equations of motion are introduced, and the DI design process is applied to a particular reentry vehicle; i.e., the X-38. Finally, several forms of desired dynamics are presented for this DI application.

3.2   Dynamic Inversion Concept (Linear Aircraft Controller)

As we suggested previously, the basic concept of DI is quite simple. In general, aircraft dynamics are expressed by

$$\dot{x} = F(x, u)$$
$$y = H(x) \tag{3.1}$$

where x is the state vector, u is the control vector, and y is the output vector. For conventional uses (where small perturbations form trim conditions), the function F is linear in u. Equation (3.1) can be rewritten as

$$\dot{x} = f(x) + g(x)u \tag{3.2}$$

where f is a nonlinear state dynamic function and g is a nonlinear control distribution function. If we assume $g(x)$ is invertible for all values of x, the control law is obtained by subtracting $f(x)$ from both sides of Equation (3.2) before multiplying both sides by $g^{-1}(x)$.

$$u = g^{-1}(x)\left[\dot{x} - f(x)\right] \tag{3.3}$$

The next step is to command the aircraft to specified states. Instead of specifying the desired states directly, we will specify the rate of the desired states, $\dot{x}$. By swapping $\dot{x}$ in the previous equation to $\dot{x}_{des}$, we get the final form of a DI control law.

$$u = g^{-1}(x)\left[\dot{x}_{des} - f(x)\right] \tag{3.4}$$

Figure 3.1 shows a block diagram representation of the DI process.
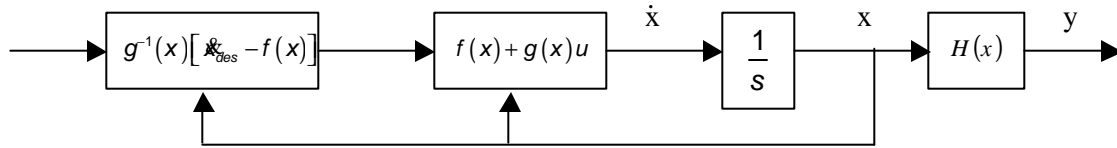
Figure 3.1 Dynamic inversion process

Although the basic DI process is simple, a few points need to be emphasized. First, although we assume $g(x)$ is invertible for all values of x, this assumption is not always true. For example, $g(x)$ is not generally invertible if there are more states than controls. Furthermore, even if $g(x)$ is invertible (i.e., $g(x)$ is small), the control inputs, u, become large; and this growth is a concern because of actuator saturation. Since the dynamics of the actuators, as well as sensor noise in the feedback loop, are neglected during this primitive controller development to illustrate the process, a "perfect" inversion is not possible.

DI is also essentially a special case of model-following. While it is similar to other model-following controllers, a DI controller requires exact knowledge of model dynamics to achieve good performance. Robustness issues therefore play a significant role during the design process. (This issue is discussed in detail in Chapter 5.) To overcome these difficulties, a DI controller is normally used as an inner-loop controller in combination with an outer-loop controller designed using other control design techniques.

The closed-loop transfer function for a desired CV that is being inverted is found according to Figure 3.2. From this block diagram, we can observe that the desired dynamics operate on the error between the commanded CV and its feedback term. In this figure, the pure integrator on the right side is used to approximate the rest of the system dynamics, as shown on the right side of the block diagram[12]. The CV here corresponds to the state x in the previous development as well as in Figure 3.1.
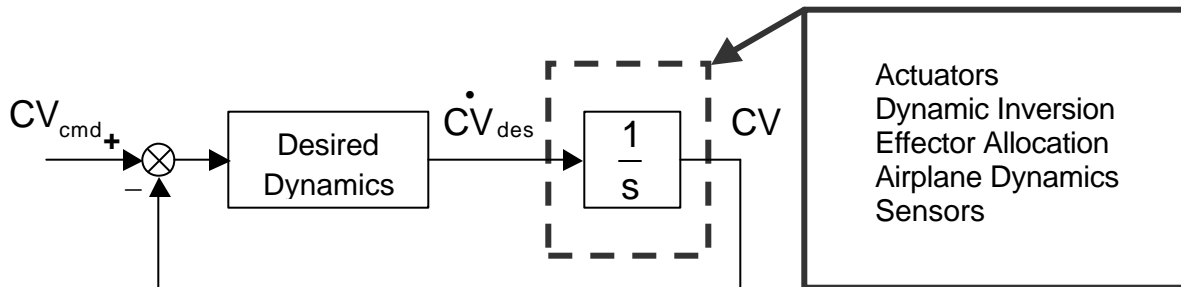
Figure 3.2 Block diagram to calculate closed-loop transfer function

3.2.1 Simplified Longitudinal Controller for an Aircraft

A simplified form of the linear longitudinal equation for an aircraft's pitch axis considers only the pitching moment equation.

$$\dot{q} = M_a a + M_q q + M_{de} de \tag{3.5}$$

11

The correspondence of this equation to Equation (3.2) is clearly seen in the following mappings:

$$x \Rightarrow q$$
$$u \Rightarrow \delta_e$$
$$f \Rightarrow M_\alpha \alpha + M_q q$$
$$g \Rightarrow M_{\delta_e}$$

Since $M_{\delta e}$ is a constant for a linear time invariant system, the inverse of the control distribution function, g, is always obtained as a constant, $1/M_{\delta e}$.

Now, we need to invert this equation for the elevator deflection angle. Mapping is achieved by substituting the relationships shown in the correspondence mappings (above) into Equation (3.3), giving the following equation:

$$\delta_e = \frac{1}{M_{\delta_e}} \left[\dot{q} - M_\alpha \alpha - M_q q \right] \tag{3.6}$$

To obtain a control law, we specify the desired value of pitch acceleration, $\dot{q}^{des}$. Then, by substituting $\dot{q}^{des}$ for $\dot{q}$ in Equation (3.6) and by substituting α^{meas} and q^{meas} for α and q, we get the following longitudinal dynamic inversion control law:

$$\delta_e^{cmd} = \frac{1}{M_{\delta_e}} \left[\dot{q}^{des} - M_\alpha \alpha^{meas} - M_q q^{meas} \right] \tag{3.7}$$

Figure 3.3 shows the block diagram representation of the longitudinal DI controller.
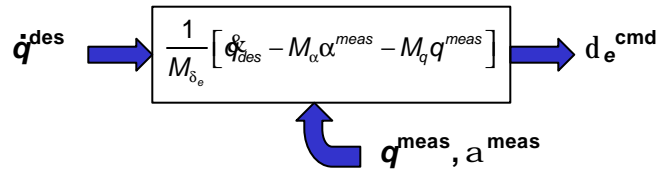
Figure 3.3 Longitudinal Dynamic Inversion Control block diagram

Recall that aircraft dynamics are modeled as a simple first-order form (Equation (3.5)) to develop this simplified DI control equation. In this model, both nonlinearity and higher-order terms in the actual aircraft dynamics are neglected. Since this simple DI controller cannot completely cancel out the aircraft dynamics, controller performance is potentially degraded.

Similarly, $\delta_e \neq \delta_e^{cmd}$ due to actuator dynamics. This shortcoming, which is also neglected while simplifying the control law development, is most noticeable when the control surface position and rate exceed their limits – something that occurs often when the value of $M_{\delta e}$ is too small (in this case, δ_e^{cmd} was unbounded).

Finally, $\alpha^{meas} \neq \alpha;$ $q^{meas} \neq q$ due to sensor processing. This factor is also neglected in the control law development, thereby potentially harming controller performance as well.

3.2.2   Simplified Lateral Directional Controller for an Aircraft

Lateral/directional DI control equations are developed in this section. Although the development procedure is similar to that of the longitudinal case, we need to simultaneously deal with two states (roll rate and yaw rate) controlled by two control surfaces (ailerons and rudders) instead of with one state (pitch rate) controlled by one control surface (elevator) as in the simplified longitudinal case.

Simplified linear lateral aircraft equations can be written with respect to roll as well as yaw axes as

$$
\begin{aligned}
\dot{p} &= L_p p + L_r r + L_\beta \beta + L_{\delta a} \delta a + L_{\delta r} \delta r \\
\dot{r} &= N_p p + N_r r + N_\beta \beta + N_{\delta a} \delta a + N_{\delta r} \delta r
\end{aligned} \tag{3.8}
$$

If we write Equation (3.8) in a compact matrix form, we get

$$
\begin{bmatrix} \dot{p} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L_p & L_r & L_\beta \\ N_p & N_r & N_\beta \end{bmatrix} \begin{bmatrix} p \\ r \\ \beta \end{bmatrix} + \begin{bmatrix} L_{\delta a} & L_{\delta r} \\ N_{\delta a} & N_{\delta r} \end{bmatrix} \begin{bmatrix} \delta a \\ \delta r \end{bmatrix}. \tag{3.9}
$$

When we compare the matrix form of Equation (3.9) to Equation (3.2), each parameter is either a vector or a matrix but the form remains the same.

$$
\begin{aligned}
\dot{\mathbf{x}} &= \begin{bmatrix} \dot{p} \\ \dot{r} \end{bmatrix} \\[6pt]
\mathbf{x} &= \begin{bmatrix} p \\ r \\ \beta \end{bmatrix} \\[6pt]
\mathbf{u} &= \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \\[6pt]
\mathbf{f} &= \begin{bmatrix} L_p & L_r & L_\beta \\ N_p & N_r & N_\beta \end{bmatrix} \\[6pt]
\mathbf{g} &= \begin{bmatrix} L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix}
\end{aligned} \tag{3.10}
$$

Notice here that the control distribution matrix, g, is a square matrix. Therefore, its inverse exists in general.

As a next step similar to the longitudinal case, we will invert the roll rate and yaw rate dynamic equations to obtain aileron and rudder deflection angles.

$$\begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} = \begin{bmatrix} L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \dot{p} \\ \dot{r} \end{bmatrix} - \begin{bmatrix} L_p & L_r & L_\beta \\ N_p & N_r & N_\beta \end{bmatrix} \begin{bmatrix} p \\ r \\ \beta \end{bmatrix} \right\} \tag{3.11}$$

Then, substituting the desired states $\dot{p}^{des}$ and $\dot{r}^{des}$ for $\dot{p}$ and $\dot{r}$ along with the measured values of p, r, and β (p^{meas}, r^{meas}, and β^{meas}) for p, r, and β, we get the lateral DI control law.

$$\begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix}^{cmd} = \begin{bmatrix} L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \dot{p} \\ \dot{r} \end{bmatrix}^{des} - \begin{bmatrix} L_p & L_r & L_\beta \\ N_p & N_r & N_\beta \end{bmatrix} \begin{bmatrix} p \\ r \\ \beta \end{bmatrix}^{meas} \right\} \tag{3.12}$$

Figure 3.4 presents a block diagram representation of the lateral DI controller.
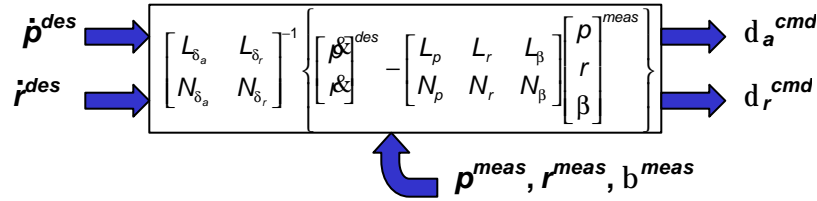
Figure 3.4 Lateral Dynamic Inversion Control block diagram

3.3   Nonlinear Dynamic Inversion

The previous examples illustrate DI control for a linear system. This approach can be readily extended to a system with nonlinear characteristics by starting with the following set of nonlinear equations typical for an aircraft.

$$\dot{p} = \frac{I_z L + I_{xz} N}{I_x I_z - I_{xz}^2} + \frac{I_{xz}\left(I_x - I_y + I_z\right)pq}{I_x I_z - I_{xz}^2} + \frac{\left[I_z\left(I_y - I_z\right) - I_{xz}^2\right]qr}{I_x I_z - I_{xz}^2}$$

$$\dot{q} = \frac{1}{I_y}\left[m + \left(I_z - I_x\right)pr + I_{xz}\left(r^2 - p^2\right)\right] \tag{3.13}$$

$$\dot{r} = \frac{I_{xz} L + I_x N}{I_x I_z - I_{xz}^2} + \frac{I_{xz}\left(I_x - I_y + I_z\right)pq}{I_x I_z - I_{xz}^2} + \frac{\left[I_x\left(I_x - I_y\right) - I_{xz}^2\right]pq}{I_x I_z - I_{xz}^2}$$

Now, we will assume the longitudinal and lateral-directional moments – L, M, and N – are linear with respect to aerodynamic derivatives; i.e.,

$$L = L_\beta \beta + L_{\delta_a} \delta_a + L_{\delta_r} \delta_r + L_p p + L_r r$$
$$M = M_\alpha \alpha + M_q q + M_{\delta_e} \delta_e \qquad . \qquad (3.14)$$
$$N = N_\beta \beta + N_{\delta_a} \delta_a + N_{\delta_r} \delta_r + N_p p + N_r r$$

By substituting the above linear moment equations into Equation (3.13), we can obtain a relation in Equation (3.15) that combines linear and nonlinear terms.

$$
\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} =
\begin{bmatrix} 0 & L_\beta & L_p & 0 & L_r \\ M_\alpha & 0 & 0 & M_q & 0 \\ 0 & N_\beta & N_p & 0 & N_r \end{bmatrix}
\begin{bmatrix} \alpha \\ \beta \\ p \\ q \\ r \end{bmatrix}
+
\begin{bmatrix} 0 & L_{\delta_a} & L_{\delta_r} \\ M_{\delta_e} & 0 & 0 \\ 0 & N_{\delta_a} & N_{\delta_r} \end{bmatrix}
\begin{bmatrix} \delta_e \\ \delta_a \\ \delta_r \end{bmatrix}
$$

$$
+
\begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{xz} & 0 & I_z \end{bmatrix}^{-1}
\begin{bmatrix} I_{xz} pq + \left(I_y - I_z \right) qr \\ I_{xz} \left(r^2 - p^2 \right) + \left(I_z - I_x \right) pr \\ -I_{xz} qr + \left(I_x - I_y \right) pq \end{bmatrix}
\qquad (3.15)
$$

If the last term is ignored, the result is identical to the linear set of DI equations previously obtained. Finally, inverting the above equation as well as performing proper substitutions of the commanded, desired, and measured values gives the resulting DI control law.

$$
\begin{bmatrix} \delta_e \\ \delta_a \\ \delta_r \end{bmatrix}^{cmd} =
\begin{bmatrix} 0 & L_{\delta_a} & L_{\delta_r} \\ M_{\delta_e} & 0 & 0 \\ 0 & N_{\delta_a} & N_{\delta_r} \end{bmatrix}^{-1}
\left\{
\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}^{des}
-
\begin{bmatrix} 0 & L_\beta & L_p & 0 & L_r \\ M_\alpha & 0 & 0 & M_q & 0 \\ 0 & N_\beta & N_p & 0 & N_r \end{bmatrix}
\begin{bmatrix} \alpha \\ \beta \\ p \\ q \\ r \end{bmatrix}^{meas}
\right.
$$

$$
\left.
-
\begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{xz} & 0 & I_z \end{bmatrix}^{-1}
\begin{bmatrix} I_{xz} p^{meas} q^{meas} + \left(I_y - I_z \right) q^{meas} r^{meas} \\ I_{xz} \left(r^{meas^2} - p^{meas^2} \right) + \left(I_z - I_x \right) p^{meas} r^{meas} \\ -I_{xz} q^{meas} r^{meas} + \left(I_x - I_y \right) p^{meas} q^{meas} \end{bmatrix}
\qquad (3.16)
$$

3.4   Applying the Dynamic Inversion Controller to the X-38 – the Overall Structure

The DI control laws developed in the previous sections are now integrated into an overall control structure. As the block diagram in Figure 3.5 shows, DI control is used as an inner loop accompanied by α and ϕ feedback outer loops. Although any type of control technique can be used for the outer loop, simple feedback is used in this particular example to illustrate the characteristics of inner-loop DI control.
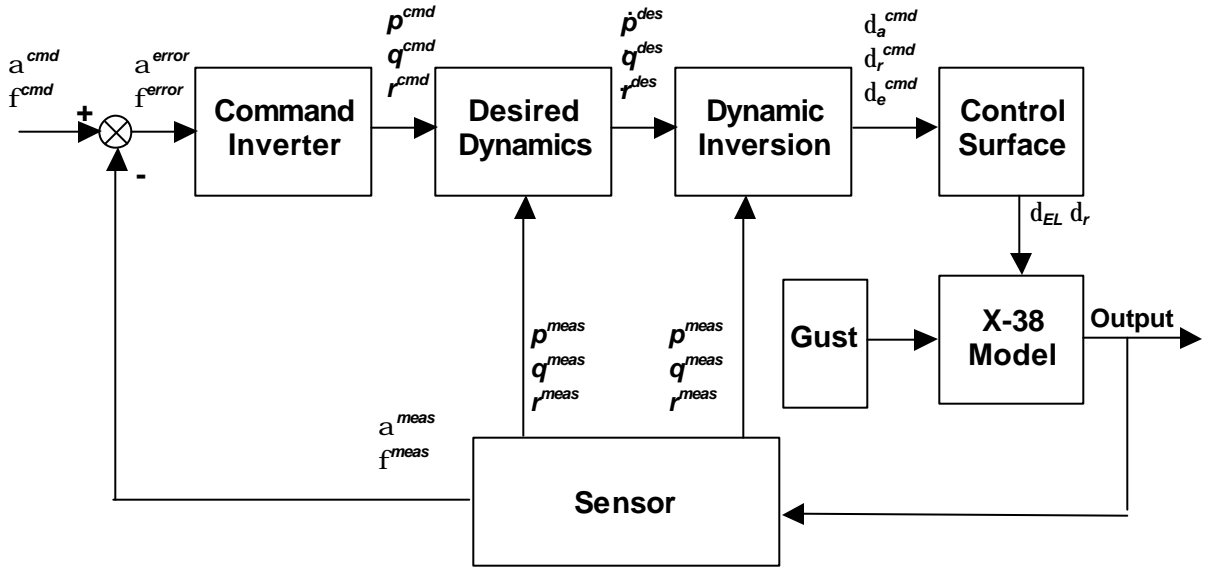
Figure 3.5 Overall Dynamic Inversion Control block diagram

The overall DI controller requires commanded values of angle-of-attack, α^{cmd}, and bank angle, ϕ^{cmd}, as inputs. Then, the measured values of α^{meas} and ϕ^{meas} are subtracted from the commanded values to produce α^{error} and ϕ^{error} in the outer loop. These error values are then fed into the Command Inverter block to be changed to rate commands, p^{cmd}, q^{cmd}, and r^{cmd}. The Desired Dynamics block uses these rate commands and the rate measurements to create the desired acceleration terms – favored forms of commands for the DI controller. The next block is the DI block, which produces the control surface deflection angle commands δ_a^{cmd}, δ_r^{cmd}, and δ_e^{cmd}. Finally, the control surface commands are fed into the Plant block, X-38 Model, via the Control Surface block. The Control Surface block includes control surface management logic, which blends the three command values, δ_a^{cmd}, δ_r^{cmd}, and δ_e^{cmd}, into two command values, δ_{EL}^{cmd} and δ_r^{cmd}, that include the dynamics of the actuators as well as the position and rate limits of the actuators. Gust and sensor noises are added to the system as external disturbances as well.

3.4.1   Command Inverter

In aircraft applications, sometimes it is better to command displacements in the angle-of-attack and bank angle rather than command the body axis rates p, q, and r. However, rate commands are needed as inputs to the Desired Dynamics block. The Command Inverter block (Figure 3.6) changes displacement commands into rate commands so that displacement commands are directly implemented in the DI controller. This section describes how displacement commands are transformed into rate commands.

16

Figure 3.6 Command Inverter block diagram

3.4.2   Body Components and Euler Angles Relationship

Roll, pitch, and yaw rates are obtained from Euler angular rates using the following transformation matrix:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} \dot\phi \\ \dot\theta \\ \dot\psi \end{bmatrix} \tag{3.17}$$

Now, by substituting the commanded values p^{cmd}, q^{cmd}, and r^{cmd} for the corresponding p, q, and r and by replacing $\dot\phi$, $\dot\theta$, and $\dot\psi$ with their corresponding commanded values, $\dot\phi^{cmd}$, $\dot\theta^{cmd}$, and $\dot\psi^{cmd} = 0$, the following relationship is obtained:

$$\begin{bmatrix} p^{cmd} \\ q^{cmd} \\ r^{cmd} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \cos\phi \\ 0 & -\sin\phi \end{bmatrix} \begin{bmatrix} \dot\phi^{cmd} \\ \dot\theta^{cmd} \end{bmatrix} \tag{3.18}$$

The next step is to express commanded values of Euler rates in terms of the commanded values of the angle-of-attack and bank angles.

3.4.3   Roll Angular Rate

The commanded roll rate, $\dot\phi^{cmd}$, is obtained from the commanded bank angle, ϕ^{cmd}, simply by differentiating with respect to time.

$$\dot\phi^{cmd} = \frac{d}{dt}\phi^{cmd} \tag{3.19}$$

By substituting the above expression into the first row of Equation (3.17), p^{cmd} is expressed as a function of ϕ^{cmd}.

$$p^{cmd} = \frac{d}{dt}\phi^{cmd} \tag{3.20}$$

3.4.4   Pitch Angular Rate

Expressing pitch angular rate, $\dot{\theta}^{cmd}$, from angle-of-attack is slightly more complicated than the roll angular rate case. First, the Euler pitch angle can be expressed in terms of α (angle-of-attack), β (sideslip angle), γ (flight path angle), and ϕ (bank angle) by

$$\theta = tan^{-1}\left(\frac{ab + sin\gamma\sqrt{a^2 - sin^2\gamma + b^2}}{a^2 - sin^2\gamma} \right),$$

(3.21)

where: $a = cos\alpha\, cos\beta$
$b = sin\phi\, sin\beta + cos\phi\, sin\alpha\, cos\beta$

The commanded value of the Euler pitch rate is calculated by differentiating the commanded value of Euler pitch angle by

$$\dot{\theta} = \frac{d}{dt}\theta.$$

(3.22)

Substituting this expression for $\dot{\theta}$ into the second row of Equation (3.17), q^{cmd} is expressed as a function of $\boldsymbol{q}^{cmd}$.

$$q^{cmd} = cos\phi\left(\frac{d}{dt}\theta^{cmd} \right)$$

(3.23)

with $\theta^{cmd} = tan^{-1}\left(\frac{a^{cmd}b^{cmd} + sin\gamma\sqrt{\left(a^{cmd}\right)^2 - sin^2\gamma + \left(b^{cmd}\right)^2}}{\left(a^{cmd}\right)^2 - sin^2\gamma} \right)$

where: $a^{cmd} = cos\,\alpha^{cmd}\,cos\beta$
$b^{cmd} = sin\phi^{cmd}\,sin\beta + cos\phi^{cmd}\,sin\alpha^{cmd}\,cos\beta$

3.4.5   Yaw Angular Rate

Instead of defining the corresponding Euler pitch and yaw rate commands to r^{cmd}, we simply set r^{cmd} equal to zero.

$$r^{cmd} = 0$$

(3.24)

3.5   Multiple Time Scale Method

To bypass a singularity problem in the inversion of an ineffective control matrix, a multiple time scale method has been developed that has been found to be quite successful in solving the problem. This approach is especially useful when inverting slow-motion variables, such as angle-of-attack, α, in the longitudinal case and sideslip, β, and bank angle, ϕ, in the lateral/directional case. These variables are deemed as "slow" dynamics because the control effectiveness on their dynamics is quite low. Variables making up the "fast" aircraft dynamics include pitch rate, q, in the longitudinal case, roll rate, p, and yaw rate, r, in the lateral/directional case. Since the control effectiveness on these body rates is high, these dynamics are considered "fast" dynamics. The multiple time scale method thus seeks to reformulate the original differential equation (Equation (3.1)) into a set of two separate differential equations consisting of a set of slow dynamics, $\dot{\mathbf{x}}$, and a set of fast dynamics, $\dot{\mathbf{y}}$.

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{y} \tag{3.25}$$

$$\dot{\mathbf{y}} = h(\mathbf{x},\mathbf{y}) + k(\mathbf{x},\mathbf{y})\mathbf{u} \tag{3.26}$$

Applying this technique to the linear aircraft dynamics, $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$, yields the following slow dynamic equations for the rate variables (Equation (3.27)) and fast dynamic equations for the acceleration variables (Equation (3.28)):

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} A_{22} & 0 & 0 \\ 0 & \overline{A}_{11} & \overline{A}_{14} \\ 0 & \overline{A}_{41} & \overline{A}_{44} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \phi \end{bmatrix} + \begin{bmatrix} 0 & A_{23} & 0 \\ \overline{A}_{12} & 0 & \overline{A}_{13} \\ \overline{A}_{42} & 0 & \overline{A}_{43} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.27}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & \overline{A}_{21} & \overline{A}_{24} & \overline{A}_{22} & 0 & \overline{A}_{23} \\ A_{32} & 0 & 0 & 0 & A_{33} & 0 \\ 0 & \overline{A}_{31} & \overline{A}_{34} & \overline{A}_{32} & 0 & \overline{A}_{33} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \phi \\ p \\ q \\ r \end{bmatrix} + \begin{bmatrix} \overline{B}_{21} & 0 & \overline{B}_{22} \\ 0 & B_{31} & 0 \\ \overline{B}_{31} & 0 & \overline{B}_{32} \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \end{bmatrix} \tag{3.28}$$

where A and B represent the longitudinal state and control input matrix values for the linear state-space model, and $\overline{A}$ and $\overline{B}$ represent the lateral/directional state and control input matrix values. Also, the subscripts denote the row and column value, respectively. Note that in Equation (3.27), rate variables form the input for the slow dynamics while the actual control surface commands form inputs for the rate dynamics shown in Equation (3.28). Inverting each set of differential equations generates two DI control laws, one for the outer DI loop (Equation (3.29)) and one for the inner DI loop (Equation (3.30)).

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix}_{cmd} = \begin{bmatrix} 0 & A_{23} & 0 \\ \overline{A}_{12} & 0 & \overline{A}_{13} \\ \overline{A}_{42} & 0 & \overline{A}_{43} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\phi} \end{bmatrix}_{des} - \begin{bmatrix} A_{22} & 0 & 0 \\ 0 & \overline{A}_{11} & \overline{A}_{14} \\ 0 & \overline{A}_{41} & \overline{A}_{44} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \phi \end{bmatrix} \right\} \tag{3.29}$$

$$\begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \end{bmatrix}_{cmd} = \begin{bmatrix} \bar{B}_{21} & 0 & \bar{B}_{22} \\ 0 & B_{31} & 0 \\ \bar{B}_{31} & 0 & \bar{B}_{32} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}_{des} - \begin{bmatrix} 0 & \bar{A}_{21} & \bar{A}_{24} & \bar{A}_{22} & 0 & \bar{A}_{23} \\ A_{32} & 0 & 0 & 0 & A_{33} & 0 \\ 0 & \bar{A}_{31} & \bar{A}_{34} & \bar{A}_{32} & 0 & \bar{A}_{33} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \phi \\ p \\ q \\ r \end{bmatrix} \right\} \qquad (3.30)$$

A block diagram representation of this 2-time scale approach is shown in Figure 3.7.
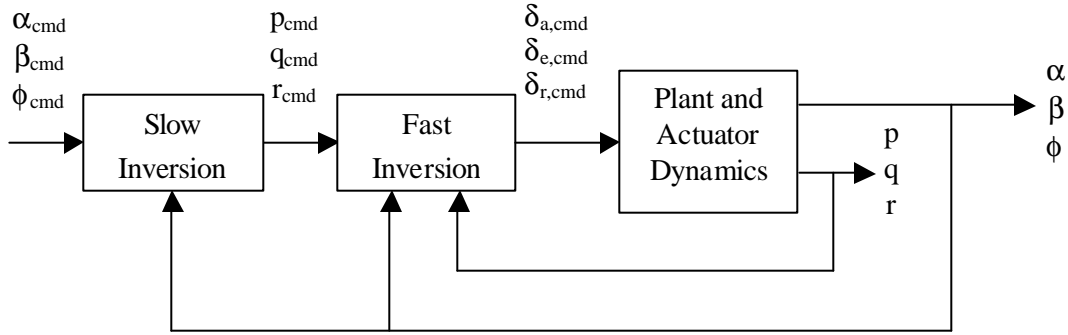
Figure 3.7 Block diagram of the 2-time scale approach

In the Fast Inversion block, fast desired dynamics are calculated and the control law in Equation (3.30) is implemented. Fast dynamics are a function of the CV commands, (p_{cmd}, q_{cmd}, and r_{cmd}) and their feedback terms (p, q, and r). Similarly, in the Slow Inversion block slow desired dynamics are calculated and the control law (Equation (3.29)) is implemented. Again, slow dynamics are a function of the CV commands (α_{cmd}, β_{cmd}, and ϕ_{cmd}) and their feedback terms (α, β, and ϕ). In summary, the Slow Inversion block produces the commanded rate variables of Equation (3.29) that are fed to the desired dynamics in the Fast Inversion block. Using these fast desired dynamics, the fast inversion control law of Equation (3.30) produces the commanded control deflections that are sent to the control surface actuators, which then serve as input to the inherent dynamics.

Several observations can be made from these two DI control laws. First, only the short-period aerodynamic terms (A_{22}, A_{23}, A_{32}, and A_{33}) are present in this set of slow and fast dynamics. Further, these two equations combine, retaining all original lateral/directional state matrix terms. It is also important to observe that the control effectiveness of the elevon on angle-of-attack, B_{21}, is not present in the inversion matrix and has actually been eliminated altogether from these two sets of equations. This is the term that traditionally causes a singularity effect on inversion because the value is typically small in magnitude. Instead, the control effectiveness on the pitch rate dynamics, B_{21}, has been retained for inversion in the fast DI control law. Similarly, control input matrix values affecting sideslip and bank angle dynamics have also been eliminated (B_{11}, B_{12}, B_{41}, and B_{42}). Therefore, only the control matrix terms for the rate dynamics have been kept (B_{21}, B_{22}, B_{31}, and B_{32}). This is of benefit because the control surfaces are more effective on the rotational rate variables than they are on the rotational variables. Finally, it is important to emphasize the fact that this 2-time scale method requires that the designer specify *two* sets of desired dynamics: one set for the slow dynamics and one set for the fast dynamics.

3.6   Desired Dynamics

The Desired Dynamics block, which was introduced during DI control law development, is explained in detail in this section.

DI control requires acceleration terms. For example, as the following longitudinal DI equation shows, a desired value of pitch angular acceleration, $\dot{q}$, is required:

$$\delta_e^{cmd} = \frac{1}{M_{\delta_e}}\left[\dot{q}^{des} - M_\alpha \alpha^{meas} - M_q q^{meas} \right] \tag{3.25}$$

However, applications normally use either displacements or rates as command states to control the system. The Desired Dynamics block acts as a mapping function between the rate commands and the desired acceleration terms, which are the required form for the DI equations. The structure of the Desired Dynamics block is shown in the flow chart in Figure 3.8.
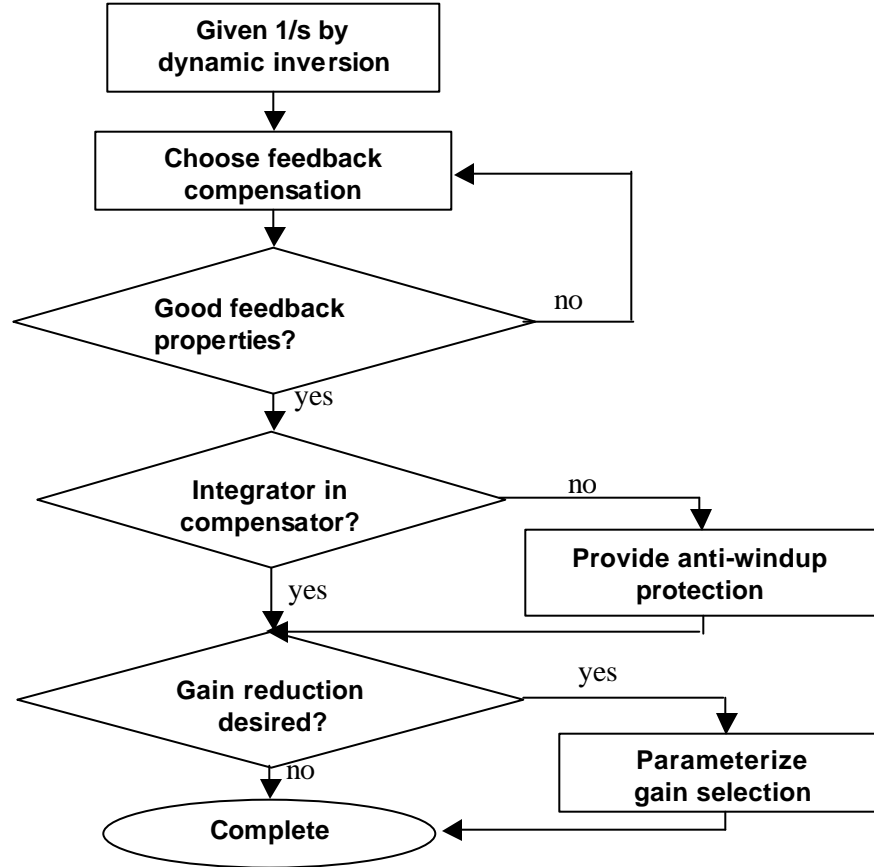
Figure 3.8 Desired dynamics development for dynamic inversion
(adapted from Ref. 12)

Several forms of desired dynamics are presented in this document and are evaluated in terms of performance and robustness. The different forms of desired dynamics consist of

- Proportional dynamics[13]
- PI dynamics[12]
- Flying quality dynamics[14]
- Ride quality dynamics

3.6.1   Proportional Case

The simplest way of achieving desired dynamics implementation is the proportional, or first-order, case. In this case, the desired dynamics are expressed as

$$\dot{CV}_{des} = K_{\omega}\left(CV_{cmd} - CV\right).$$

(3.31)

The K_{ω} term in Equation (3.31) sets the bandwidth of the response. The bandwidth must be selected to satisfy time-scale separation assumptions without exciting structural modes or becoming subject to the rate limiting of the control actuators. Figure 3.9 shows the block diagram representation of the Proportional Desired Dynamics block introduced in this section.
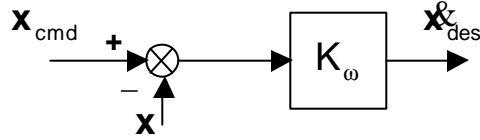
Figure 3.9 Proportional Desired Dynamics block diagram

As shown above, the constant K_{ω} amplifies the error between the CV command and its feedback term. In Figure 3.9, CV is represented as the state, **x**. So, the closed-loop transfer function for the proportional form of desired dynamics, shown in Equation (3.32), desires to place a single pole at $s = -K_{\omega}$.

$$\frac{CV}{CV_{cmd}} = \frac{K_{\omega}}{s + K_{\omega}}$$

(3.32)

3.6.2   Proportional Integral Case

The Desired Dynamics block is not limited to a first-order component. If the Desired Dynamics block does not create satisfactory handling qualities (for piloted aircraft) using a set of first-order equations, a higher-order system is used. A commonly used higher-order block is a PI. This form is particularly popular in DI literature that uses fighter aircraft examples[12,15]. This type of Desired Dynamics block structure is also used in the linearized MACH controller designed by Honeywell for the X-38 vehicle and has been adopted for this study as well. The block diagram representation of a PI desired dynamics component is shown in Figure 3.10. It has the same form as that used in the Honeywell study[12] with a K_B of 5 sec^{-1} selected.
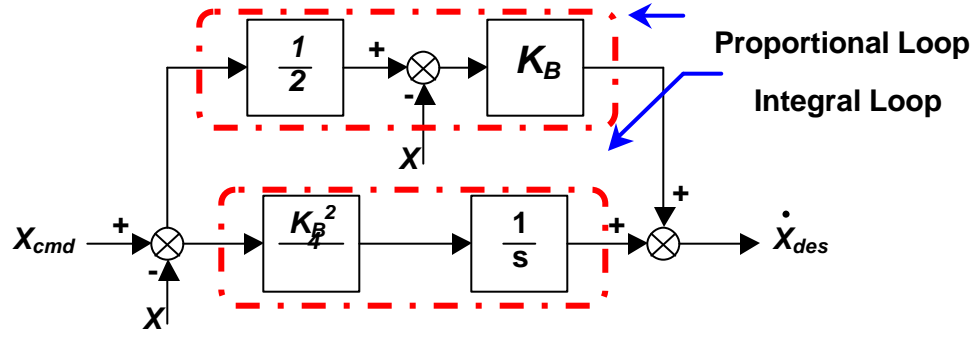
Figure 3.10 Proportional Integral Desired Dynamics block diagram

The block diagram for the PI form corresponds to the desired dynamics in Equation (3.33), where **x** is the CV.

$$\dot{CV}_{des} = K_B\left(\frac{1}{2}CV_{cmd} - CV\right) + \frac{K_B^2}{4s}\left(CV_{cmd} - CV\right) \tag{3.33}$$

These dynamics have a closed-loop transfer function of

$$\frac{CV}{CV_{cmd}} = \frac{\frac{1}{2}K_B}{s + \frac{1}{2}K_B}. \tag{3.34}$$

which seeks to place a pole at $s = -0.5K_B$ for any real constant K_B. In essence, this form of dynamics compensates for both the CV and the CV rate.

3.6.3   Flying Qualities Case

Desired dynamics can also be specified in terms of flying quality levels. Mil-STD-1797A[14] contains the flying quality specifications for different vehicle classes and mission types. Using this information, the proper time domain characteristics corresponding to a desired flying quality level (damping ratio, natural frequency, time constant) can be selected. These characteristics can be used to determine the proper values for the gains and poles locations in this form of desired dynamics. The flying qualities desired dynamics, shown in Figure 3.11, can be represented as

$$\dot{CV}_{des} = \frac{K_{fq}(s+a)}{s^2 + bs + c}\left(CV_{cmd} - CV\right), \tag{3.35}$$

where $b = 2\zeta_{des}\omega_{n,des}$ and $c = \omega_{n,des}^2 - K_{fq}$ for the desired damping ς_{des} and natural frequency $\omega_{n,des}$. Both the gain, K_{fq}, and zero location, a, are real constant values.
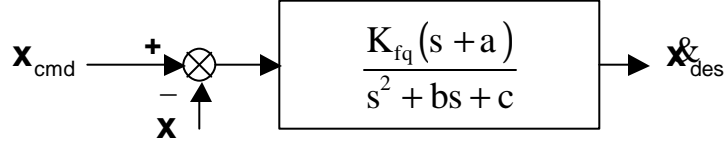
Figure 3.11 Flying Qualities Desired Dynamics block diagram

The closed-loop transfer function for the flying qualities dynamics is given in Equation (3.35). These dynamics desire to place three closed-loop poles and to add a single zero to the system.

$$\frac{CV}{CV_{cmd}} = \frac{K_{fq}(s+a)}{s^3 + bs^2 + (c + K_{fq})s + K_{fq}a} \tag{3.36}$$

3.6.4   Ride Qualities Case

The ride qualities form of desired dynamics that can also be used in DI are given in Equation (3.37) and represented pictorially in Figure 3.12.

$$\overset{g}{CV}_{des} = \frac{K_{rq}}{s+b}(CV_{cmd} - CV) \tag{3.37}$$
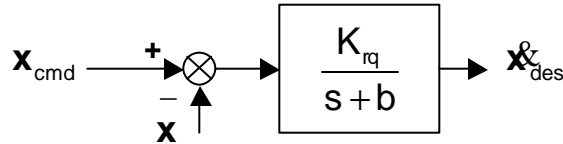
Figure 3.12 Ride Qualities Desired Dynamics block diagram

The closed-loop transfer function for this set of desired dynamics is given below by

$$\frac{CV}{CV_{cmd}} = \frac{K_{rq}}{s^2 + bs + K_{rq}}, \tag{3.38}$$

which desires two places to closed-loop poles at $s_{1,2} = -0.5b \pm 0.5\sqrt{b^2 - 4K_{rq}}i$ for real constants b and K_{rq}.

For highly augmented airplanes, the control anticipation parameter (CAP) replaces the longitudinal short period requirements, such as damping ratio and natural frequency[14]. The desired longitudinal dynamics are instead designed by selecting a desired damping ratio and CAP value. Equation (3.39) shows how to calculate CAP for a specific load factor n_α and natural frequency.

$$CAP = \frac{\omega_n^2}{n_\alpha} \tag{3.39}$$

The flying quality levels for various CAP and ζ_{SP} values are shown in Figure 3.13.
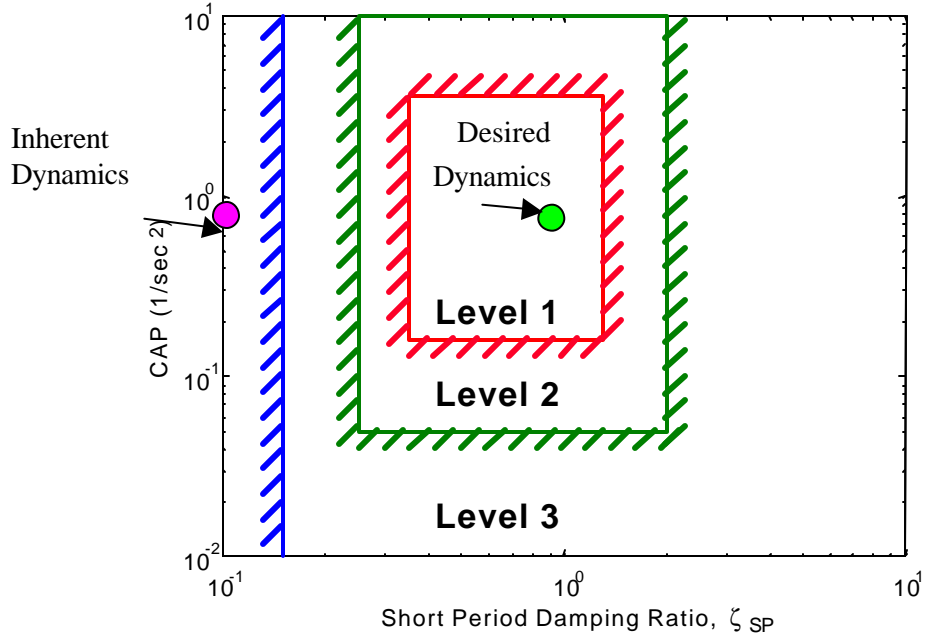
Figure 3.13 Control anticipation parameter requirements for highly augmented vehicle

Once a CAP and a ζ_{SP} are selected to satisfy a desired flying quality level, the desired short-period natural frequency can be calculated from Equation (3.39). The gain and the pole location for the open-loop desired dynamics are then backed out from these two specifications, ζ_{SP} and $\omega_{n,SP}$. For example, the desired dynamics (see Figure 3.13) correspond to the following:

- CAP = 0.802
- $\zeta_{SP} = 0.8$

Using the CAP equation yields a desired $\omega_{n,SP}$ of $1.4s^{-1}$. The corresponding gain and pole locations are then found according to Equations (3.40) and (3.41), respectively.

$$K_{rq} = \omega^2_{n,SP} = 1.96 \tag{3.40}$$

$$-b = -2\zeta_{SP}\sqrt{K_{rq}} = -2\,\zeta_{SP}\omega_{n,SP} = -2.24 \tag{3.41}$$

The desired dynamics for this example become

$$C\overset{g}{V}_{des} = \frac{1.96}{s+2.24}\left(CV_{cmd} - CV\right). \tag{3.42}$$

3.7 Issues in Dynamic Inversion

The procedure illustrating the main steps in DI controller design is shown below. Also listed next to each step are some of the issues that were encountered when first learning and using this design methodology. Some possible solutions, or options, to each of these issues are listed below the procedure.

STEPS

1. Select dynamic equation(s) to replace.

2. Select desired dynamics.

3. Form the control law.

4. Design DI inner loop(s).

5. Design outer robustness loop.

ISSUES

Does the inverse of the control input matrix, B^{-1} or $g(\mathbf{x})^{-1}$, exist?
If so, is it close to singularity?

What form should they take?

What happens when the control law asks too much of the control effectors?

If a 2-time scale approach is used, how are the two loops designed?

Is this really needed?
What type do you use?
If pole placement is used, where are the outer-loop poles placed?

Options:

1. If the inverse of the control input matrix does not exist, a multiple time scale method can be used – such as the 2-time scale approach presented in Section 3.5. A command inverter, such as those presented in Section 3.4, or a higher order of feedback linearization is also a possibility.

2. There are no limitations on the form the desired dynamics may take. However, some of the common forms found in the literature include proportional, PI, and flying qualities.

3. If redundant control effectors are available, a control allocation scheme can be designed in an effort to keep the required control deflections within the constraints of the actuator. Adjustment or replacement of the desired dynamics may also help reduce the control response.

4. A robust outer loop is required because DI alone does not guarantee robustness. The most popular robust outer-loop design methodology for DI controllers is μ-synthesis. Although in

the literature LQG is not as well published in regard to DI controllers, LQG has been shown to be effective, as shown in Section 5.2. LQR is another possibility for robust outer-loop design as well.

4   Simulation

In this chapter, the simulation component buildup is presented for the X-38 reentry vehicle, and the DI design process is then applied to the vehicle model. Simulation results are shown and discussed for three DI design examples to give the reader more detail on how the design process is carried out.

4.1   Control Surfaces

The Control Surfaces block (Figure 4.1) is composed of two sub-blocks, the Control Surface Management Logic sub-block and the Actuator Dynamics sub-block. A conventional aircraft is usually equipped with three control surfaces; namely, a rudder, an elevator, and an aileron. But, the X-38 has only two sets of control surfaces – rudders and elevons. Control Surface Management Logic must therefore blend three inputs into two inputs so that commands can be fed smoothly into the Actuator Dynamics sub-block.
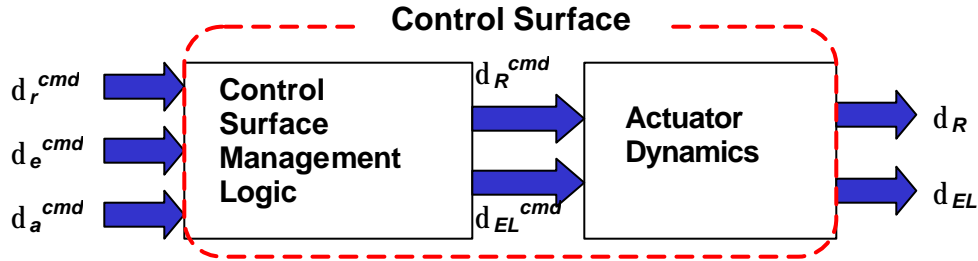
Figure 4.1 Control Surfaces block diagram

4.1.1   Definitions (from the X-38 Aerodynamic Design Data Book)

The X-38 vehicle has two sets of control surfaces (see Figure 4.2): a pair of elevon control surfaces, located on the lower rear of the vehicle; and a pair of rudders, one at the top of each of the vertical fins. The positive direction of deflection for the elevon is down (as shown in Figure 4.2), and, looking from the rear, to the left for the rudders. Both elevons and rudders are dual-function control surfaces. Each surface is deflected independently to provide the required control authorities.
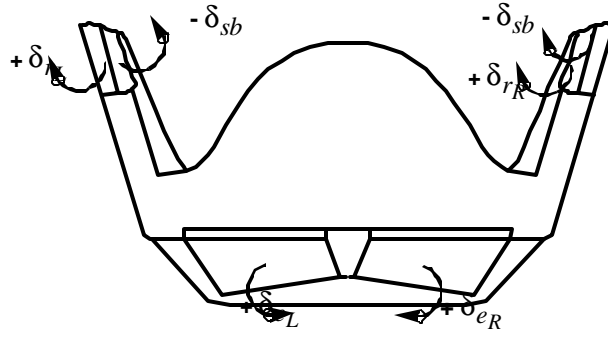
Figure 4.2 Control surface deflections

The elevon deflections are averaged to give the total elevon angle or elevator angle for pitch control.

$$\delta_e = \frac{\left(\delta_{eL} + \delta_{eR}\right)}{2} \tag{4.1}$$

The average of the difference gives aileron angles for roll control.

$$\delta_a = \frac{\left(\delta_{eL} - \delta_{eR}\right)}{2} \tag{4.2}$$

Similarly, the rudders' deflections are averaged to give total rudder for yaw control.

$$\delta_r = \frac{\left(\delta_{rL} + \delta_{rR}\right)}{2} \tag{4.3}$$

The average of the difference gives speed brake angles for pitch and drag control.

$$\delta_{sb} = \frac{\left(\delta_{rL} - \delta_{rR}\right)}{2} \tag{4.4}$$

4.1.2   Control Surface Limits

4.1.2.1   Deflection position limit

The X-38 control surface deflection limits are listed in Table 4.1.

Table 4.1 X-38 Control Surface Deflection Limits

Bodyflap Lower Deflection Limit	0.0º
Bodyflap Upper Deflection Limit	45.0º
Rudder Lower Deflection Limit	-25.0º
Rudder Upper Deflection Limit	25.0º

4.1.2.2   Surface actuator rate limits

Though the actual rate limits of the actuators used for the X-38 vehicle are set as a function of hinge moment, the constant values shown in Table 4.2 were used for this study.

Table 4.2 X-38 Control Surface Rate Limits

Body Flap Deflection Rate Limit	50 deg/sec
Rudder Deflection Rate Limit	50 deg/sec

4.1.3   Control Actuator Modeling

Control surface actuators are modeled with the following second-order lag for both rudders and body flaps:

$$G_{ACTUATOR}(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \qquad (4.5)$$

with ω_n = 26 rad/sec and ζ = 0.707 in both sets of actuators.

4.1.4   Control Surface Management

Unlike conventional aircraft, the X-38 is equipped with only two pairs of control surfaces; namely, rudders and elevons. By commanding the deflections either symmetrically or asymmetrically, these two pairs of surfaces provide the same control effects that conventional rudders, elevators, ailerons, and speed brakes provide. Currently, the speed brake mode is turned off for our model.

Three different command signals, which must be converted into excitation signals for the two pairs of control surfaces, come from the DI controller. The following flow chart (Figure 4.3) illustrates the methodology used to blend the control surface movements. Basically, this control surface management logic assigns priority to pitch axis control over the other two axes.
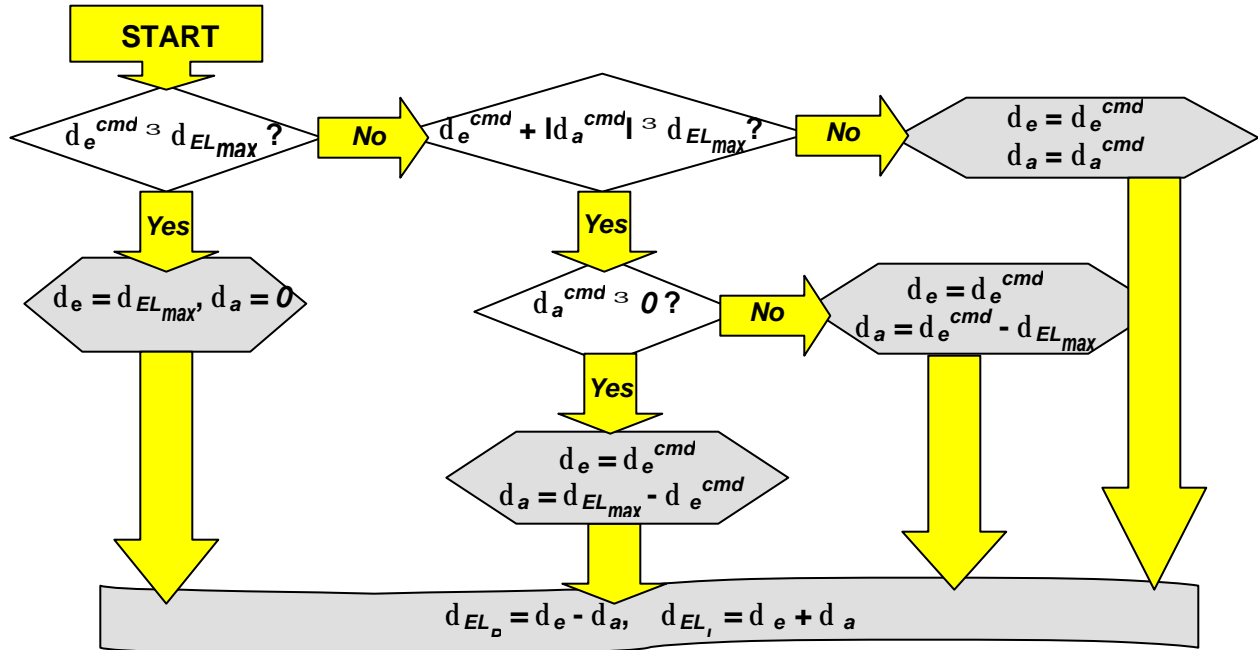
Figure 4.3 Elevon control management logic flow chart

Since optimization of the control surface allocation is beyond the scope of this study, a very simple surface management logic is developed and employed here. However, this allocation is an important aspect of the ultimate design of the flight control system. An important question yet to be answered is whether the DI design process flows naturally into this optimization scheme, or whether it makes optimization more obscure. Figures 4.3 (above) and 4.4 (below) illustrate control surface management logic for rudders and elevons.
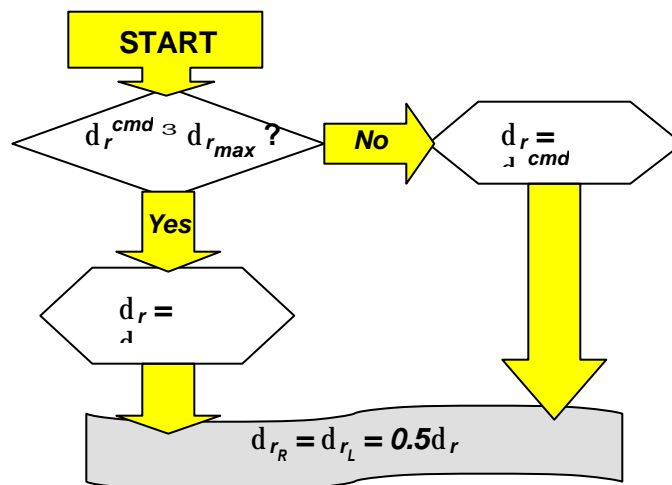
Figure 4.4 Rudder control management logic flow chart

4.2 Sensor Modeling

As in other aircraft, direct measurement of sideslip angle on the X-38 is highly uncertain. So, the sideslip angle is estimated by combining other measurements. The following equation is one way to estimate sideslip angle. An integral form of this equation is used in simulations for this study.

$$\dot{\beta}^{est} = \frac{Y_p}{V_T^{meas}}p + \left(\frac{Y_r}{V_T^{meas}} - 1\right)r^{meas} + \left(\frac{g}{V_T^{meas}}\right)\phi^{meas} + \left(\frac{Y_{\delta r}}{V_T^{meas}}\right)\delta r^{meas} \qquad (4.6)$$

All other parameters are assumed to be measurable. White noise is added to the outputs to mimic sensor noise.

4.3 Gust Modeling

Gust effects, modeled as a disturbance, are also considered when evaluating the DI controller. Gusts are added to the body component velocity of the vehicle as shown in the following block diagram (Figure 4.5). Typical gust inputs are also shown in Figure 4.6.
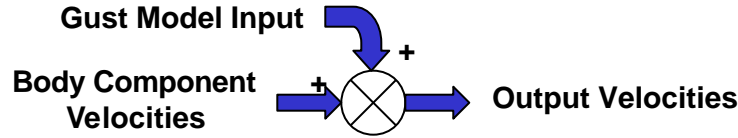
Figure 4.5 Gust modeling
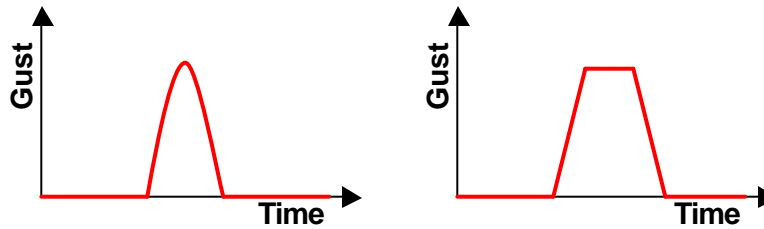
Figure 4.6 Typical gust inputs

4.4 Comparison Between MACH Controller and TAMU Design

4.4.1 Control Variable Definition

Derivatives of the state variables p, q, and r are used as CVs for the X-38 DI example. Instead of specifying p, q, or r, however, MACH specifies the dynamics of the derivative of the CVs. The CV vector has three components: *LCV*, *MCV*, and *NCV*. Each of these components controls roll, pitch, and yaw moments, respectively. The variables are defined as follows:

$$CV = \begin{bmatrix} LCV \\ MCV \\ NCV \end{bmatrix} \tag{4.7}$$

with

$$LCV = p_s \tag{4.8}$$

$$MCV = q + \frac{g}{V_{co}} \frac{\lambda_{nz}}{(s+\lambda_{nz})} nz + \frac{g}{V cos\beta} cos\gamma cos\mu - p_s tan\beta + K_{\bar{q}}\bar{q} - cos\gamma\left(\frac{g}{V} + \frac{g}{V_{co}}\right) \tag{4.9}$$

$$NCV = r_s + K_\beta\hat{\beta} - \frac{g}{V} cos\gamma sin\mu \tag{4.10}$$

In these equations, μ represents the bank angle about the velocity vector.

Since the CVs are no longer states themselves but are combinations of states, a CV definition block was added to our Simulink example to "upgrade" to the MACH-generated controller. Although differences in roll axis CVs are minor, they can occur in either body-axis roll rates with flight path components or in stability axes. And although the definitions in pitch and yaw axes CVs for the MACH controller differ from this example, both have a strong angular rate content. Because the first terms on the right side of each equation dominate, *MCV* and *NCV* essentially become q and r, respectively, in our example controller.

4.4.2 Desired Dynamics Module

The Desired Dynamics module of the MACH controller is, in form, a PI controller that is identical to our Simulink example in Figure 3.10. It has the same magnitude of bandwidth, K_B, of 5 sec^{-1} as well.

4.4.3 Dynamic Inversion

The MACH controller starts with the same nonlinear form for the vehicle equations of motion as outlined in Section 3.2.

$$\dot{x} = F(u,x) \\ \approx f(x) + g(x)u \tag{4.11}$$

where $\boldsymbol{x}$ is the vector of state variables and $\boldsymbol{u}$ is the vector of control effectors. Recalling that the CVs in MACH are functions of the state variables,

$$\boldsymbol{CV} = \boldsymbol{CV}(\boldsymbol{x}), \tag{4.12}$$

where *CV* is the CV vector defined in Equation (4.7). Then,

$$\dot{\boldsymbol{CV}} = \frac{\partial \boldsymbol{CV}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = \frac{\partial \boldsymbol{CV}}{\partial \boldsymbol{x}}f(\boldsymbol{x}) + \frac{\partial \boldsymbol{CV}}{\partial \boldsymbol{x}}\boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}. \tag{4.13}$$

So, the DI control law can be obtained as

$$u^{cmd} = \left[\frac{\partial \boldsymbol{CV}}{\partial x}g(x)\right]^{-1}\left[\dot{CV}^{des} - \frac{\partial \boldsymbol{CV}}{\partial x}f(x)\right]. \tag{4.14}$$

4.4.4   Control Effector Priority (Surface Management)

As was previously explained in Section 3.2, the first bracket in the right side of the DI control law must be a square matrix so it can be inverted.   This mathematical requirement means that the number of control effectors must equal the number of CVs. In the MACH application, the row dimension of $\dfrac{\partial \mathbf{CV}}{\partial \mathbf{x}}$ is 3; therefore, the column dimension of $\boldsymbol{g}(\boldsymbol{x})$ must be 3 as well to have a square matrix. Thus, three control effectors are required. Since the three CVs in MACH have a strong angular rate content and the X-38 has three moment-producing controls (differential elevons, symmetric elevons, and rudders), the vehicle has an appropriate number of control effectors for three of the four flight control modes listed in Table 4.3.

Table 4.3 MACH V201 Flight Control Modes

Mode	Flight Mode	Conditions	# of CVs	CV Feedback	# of Effectors	Effectors
1	ACS Only	$q_{bar} < 2$	3	INS: p, q, r, β	3	T_x, T_y, T_z
2	Blended ACS	$2 < q_{bar} < 30$	3	INS: p, q, r, β, N_z NAVDAD: α, V, q_{bar}	5	$T_x, T_y, T_z, \delta_a, \delta_e$
3	Flaps Only	$q_{bar} > 30$ and $M < 6$	2	INS: p, q, r, N_z NAVDAD: α, V, q_{bar}	2	δ_a, δ_e
4	Rudders and Flaps	$M > 6$	3	INS: p, q, r, N_z, N_y NAVDAD (or FADS): α, V, q_{bar}	3	$\delta_a, \delta_e, \delta_r$

Note: T_x, T_y, *and* T_z *are the reaction control thrusters, and* **d_a**, **d_e**, *and* **d_r** *are the resulting effective control surface deflections.*

The four flight control modes presented in Table 4.3 fit two different situations (for the MACH controller) as follows:

1. The number of controlled variables equals the number of available control effectors (modes 1, 3, and 4).
2. The number of controlled variables is less than the number of available control effectors (mode 2).

In mode 3, there are only two available control effectors; the rudders are fixed. Consequently, the number of controlled variables is greater than the number of available control effectors. To overcome this problem (having more unknowns than equations), the number of controlled variables is reduced to two by combining roll and yaw signals. Then, the DI control law can be obtained.

When redundant actuators exist (mode 2), a unique solution for the actuator commands can be found by effectively reducing the number of available controls. MACH applies a weighted and biased pseudo-inverse solution. Details of this approach are spelled out in the literature[12].

The MACH controller also incorporates an algorithm called *sum* that deals with all possible combinations of the number of unknowns and the number of equations described in this section. The *sum* algorithm is described in detail in Appendix A of Reference 10.

4.4.5   Least-Squares Aerodynamic Model

The Simulink-based DI control examples presented in the following sections use time-invariant coefficients. Unfortunately, aerodynamic coefficients vary over time for all practical applications. All coefficients in the DI controller must therefore be updated to account for flight conditions in which the vehicle is operating, and how these conditions change the aerodynamic coefficients. However, it is not practical to carry all aerodynamic data at every individual point throughout the entire flight envelope. For purposes of making appropriate updates, least-squares curve-fitting functions are applied to the aerodynamic data *a priori*. This curve-fitting function is also carried on board the aircraft to generate the required coefficient updates. The six aerodynamic coefficients used in V132 MACH modeling are functions of the following variables:

Longitudinal: $\hat{C}_i = K_1(\alpha)\dfrac{\bar{c}}{2V}q + K_2(\alpha)\delta_e + K_3(\alpha), \qquad i = D,L,m$

Lateral: $\hat{C}_i = K_4(\alpha)\beta + K_5(\alpha)\dfrac{b}{2V}p + K_6(\alpha)\dfrac{b}{2V}r + K_7(\alpha)\delta_a + K_8(\alpha)\delta_r, \quad i = Y,l,n$

All Ks are a function of α only, and the Mach number is held constant at 0.6 in the described MACH controller.

When the same set of coefficients is applied to V201 in the TAMU example controller, the least-squares curve fitting is modified to be nonlinear both in α and in the Mach number for longitudinal and lateral/directional coefficients.

Lateral/directional derivatives are formed using an elevon trim angle setting, δ_e, based on a linear relationship of α for longitudinal trim over the Mach range.

Longitudinal: $\hat{C}_k = C_k(\alpha,M) + C_{k_q}(\alpha,M)\dfrac{q\bar{c}}{2V} + C_{k_{\delta_e}}(\alpha,M)\delta_e, \quad k = D, L, m$

Lateral: $\hat{C}_k = C_{k_\beta}(\alpha,M,\delta_e)\beta + C_{k_p}(\alpha,M,\delta_e)\dfrac{pb}{2V} + C_{k_r}(\alpha,M,\delta_e)\dfrac{rb}{2V}$

$$+ C_{k_{\delta_a}}(\alpha,M,\delta_e)\delta_a + C_{k_{\delta_r}}(\alpha,M,\delta_e)\delta_r, \qquad k = Y, l, n$$

with $\delta_e = m(M)\alpha + b(M)$

4.4.6   Outer Loops

4.4.6.1   Bank angle outer loop

Block diagrams of the bank angle outer loop used in the Simulink example and MACH controllers are shown in Figure 4.7.
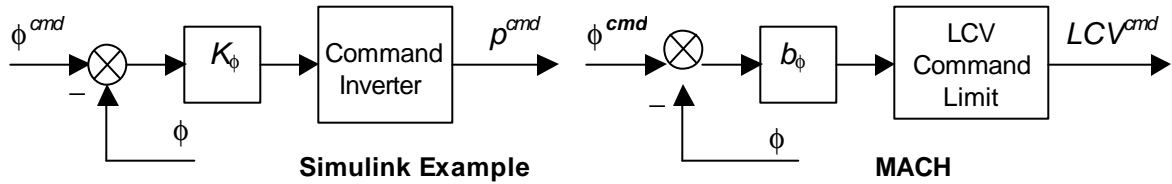
Figure 4.7 Comparison of roll angle outer loop structure

Obviously, both controllers have the same structure. Also, the outer-loop gains, K_ϕ and b_ϕ, are both normally set to 1 sec^{-1} before they are tuned to achieve desired performance.

The p^{des} signal is passed through *LCV* Command Limit block for the MACH controller. This limiter consists of

1. an absolute command limit based on JSC specifications,
2. a command limit that prevents uncontrollable inertial coupling into the pitch axis,
3. a command limit that prevents uncontrollable inertial coupling into the yaw axis, and
4. an absolute command minimum that allows some commands to get through if the inertial coupling limits go to zero.

More details of the Command Limit block in the MACH controller can be found on pages 4 and 5 of Reference 10.

4.4.6.2   Angle-of-attack outer loop

The alpha outer loops for the Simulink example and the MACH controller also have essentially the same structure (Figure 4.8). Both have unity feedback gain ($K_\alpha = b_\alpha = 1$). Once again, slight differences in the two controllers come from definitions in the command variables ($MCV \neq q$).
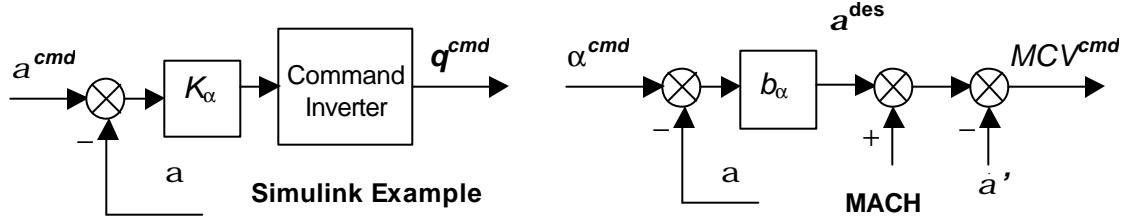
Figure 4.8 Comparison of angle-of-attack outer loop structure

Finally, in terms of outer-loop structure, neither controller contains an outer loop that will control yaw angle.

4.4.7 Comparison of Aircraft Models

The following nonlinear model[12] is used for the MACH controller:

$$I_{xx}\dot{p} - I_{xz}\dot{r} = l_a^c + l_p^c + I_{yy}rq + I_{xz}pq - I_{zz}qr$$

$$I_{yy}\dot{q} = m_a^c + m_p^c - I_{xz}p^2 + I_{zz}pr - I_{xx}rp + I_{xz}r^2$$

$$-I_{xz}\dot{p} + I_{zz}\dot{r} = n_a^c + n_p^c + I_{xx}pq - I_{xz}qr - I_{yy}pq$$

$$\dot{V} = \frac{1}{m}\left(-D\cos\beta + Y\sin\beta + T\cos\beta\cos\alpha\right) - g\sin\gamma$$

$$\dot{\chi} = \frac{1}{mV\cos\gamma}\left[D\sin\beta\cos\mu + Y\cos\mu\cos\beta + L\sin\mu + T\left(\sin\mu\sin\alpha - \cos\mu\sin\beta\cos\alpha\right)\right]$$

$$\dot{\gamma} = \frac{1}{mV}\left[-D\sin\beta\sin\mu - Y\sin\mu\cos\beta + L\cos\mu + T\left(\cos\mu\sin\alpha + \sin\mu\sin\beta\cos\alpha\right)\right] - \frac{g\cos\gamma}{V}$$

$$\dot{\mu} = \frac{p\cos\alpha + r\sin\alpha}{\cos\beta} + \frac{1}{mV}\left[D\sin\beta\cos\mu\tan\gamma + Y\tan\gamma\cos\mu\cos\beta + L\left(\tan\beta + \tan\gamma\sin\mu\right)\right.$$
$$\left. + T\left(\sin\alpha\tan\gamma\sin\mu + \sin\alpha\tan\beta - \cos\alpha\tan\gamma\cos\mu\sin\beta\right)\right] - \frac{g\cos\gamma\cos\mu\tan\beta}{V}$$

$$\dot{\alpha} = q - \tan\beta\left(p\cos\alpha + r\sin\alpha\right) - \frac{1}{mV\cos\beta}\left(L + T\sin\alpha\right) + \frac{g\cos\gamma\cos\mu}{V\cos\beta}$$

$$\dot{\beta} = -r\cos\alpha + p\sin\alpha + \frac{1}{mV}\left(D\sin\beta + Y\cos\beta - T\sin\beta\cos\alpha\right) + \frac{g\cos\gamma\sin\mu}{V}$$

$$\dot{\xi} = V\cos\gamma\cos\chi$$
$$\dot{\eta} = V\cos\gamma\sin\chi$$
$$\dot{h} = V\sin\gamma$$

with

$$l_a^c = \bar{q}SbC_l$$

$$m_a^c = \bar{q}SbC_m$$

$$n_a^c = \bar{q}SbC_n$$

4.4.8   Sensor Processing

The MACH controller assumes full-state feedback. However, since the air data system for the X-38 vehicle does not provide information regarding sideslip angle β, this angle must be estimated. Figure 4.9 shows the block diagram of the sideslip estimation block in the MACH controller.
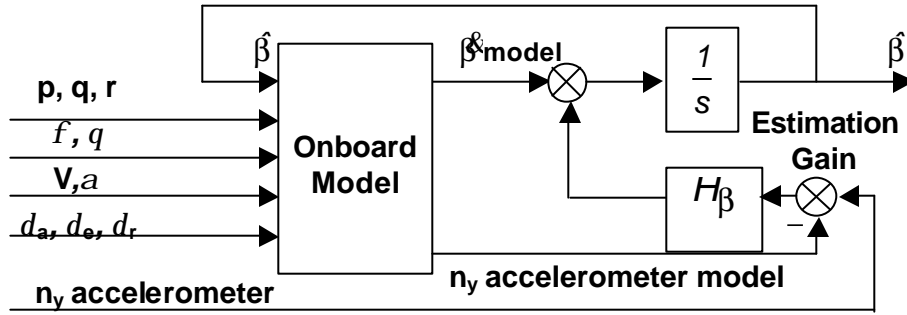
Figure 4.9 Sideslip Estimation block diagram (MACH controller)

4.5   X-38 Mathematical Model

4.5.1   Overview and Vehicle Parameters

X-38 vehicle coefficients were extracted from the output of a modified version of the SES[16]. The subroutine X35_AERO_DERIVS calculates aerodynamics coefficients periodically along the vehicle trajectory at user-specified sampling intervals. Since this subroutine does not provide trimmed values of coefficients, for this study a trim routine was not added to the program. The coefficients used in the simulation are instead taken from untrimmed flight conditions.

Table 4.4 shows some of the parameters of the X-38 V201 vehicle used during the SES simulation as well as the DI controller simulation that will be described in the next section.

Table 4.4. Mass Properties and Geometry for the X-38

Parameter	Value	Units
Weight	17578	lbs
x-axis inertia, I_x	8090	slugs/ft^2
y-axis inertia, I_y	25900	slugs/ft^2
z-axis inertia, I_z	29200	slugs/ft^2
x-z product of inertia, I_{xz}	1300	slugs/ft^2
Reference area, S	260	ft^2
Reference span, b	27.5	ft
Reference cord, $\bar{c}$	10.8	ft

4.5.2   X-38 Equations of Motion

The following nonlinear set of equations is used and integrated in the simulation:

$$\dot{p} = L_b\,\boldsymbol{b} + L_p\,p + L_r\,r + L_{da}\,\boldsymbol{da} + L_{dr}\,\boldsymbol{dr}$$

$$\dot{r} = N_b\,\boldsymbol{b} + N_p\,p + N_r\,r + N_{da}\,\boldsymbol{da} + N_{dr}\,\boldsymbol{dr}$$

$$\dot{q} = M_a\,\boldsymbol{a} + M_q\,q + M_{de}\,\boldsymbol{de}$$

$$\dot{b} = \frac{Y_b}{V_T}\,\boldsymbol{b} + \frac{Y_p}{V_T}\,p + \left(\frac{Y_r}{V_T}-1\right)r + \frac{g}{V_T}\,\boldsymbol{f} + \frac{Y_{dr}}{V_T}\,\boldsymbol{dr}$$

$$\dot{f} = p + (\sin f)(\tan q)q + (\cos f)(\tan q)r$$

$$\dot{q} = (\cos f)q - (\sin f)r$$

$$\dot{y} = (\sin f)(\sec q)q + (\cos f)(\sec q)r$$

$$\dot{V}_T = X_v V_T + X_a\,\boldsymbol{a} - g(\cos g)q + X_{de}\,\boldsymbol{de}$$

$$\dot{a} = \frac{Z_v}{V_T}V_T + \frac{Z_a}{V_T}\,\boldsymbol{a} - \frac{g(\sin g)}{V_T}q + \left(1+\frac{Z_q}{V_T}\right)q + \frac{Z_{de}}{V_T}\,\boldsymbol{de}$$

$$\dot{g} = \frac{g\cos g}{V_T}(\cos f - 1)$$

$$\dot{h} = V_T \sin g \tag{4.15}$$

4.6   Design Example 1

This section shows the simulation results obtained using the DI controller developed in Sections 3.4, 3.5, 4.1, and 4.2 of this document for the X-38 vehicle. The structure of the simulation, as shown in Figure 3.5, is built up in a MATLAB/Simulink environment. A total of ten different simulation cases is presented for three different flight conditions. The objective of this simulation exercise is to illustrate the characteristics of the DI controller as applied to the X-38 lifting-body reentry vehicle with its rather large flight envelope.

4.6.1 Flight Conditions

Three flight conditions along the nominal trajectory of the X-38 are selected and used as initial conditions of the simulations to evaluate the performance of the designed controller. Those flight conditions are selected to represent three different regimes of the flight envelope – supersonic, transonic, and subsonic. Table 4.5 summarizes the fight conditions evaluated. Since the SES does not provide trimmed values of the vehicle aerodynamic coefficients, pitch, roll, and yaw coefficients are nonzero values.

Table 4.5 Summary of Evaluated Flight Conditions

	H_0 (kft)	$M_{\infty 0}$	$\bar{q}_0$ (psf)	α_0 (°)	β_0 (°)	ϕ_0 (°)	p_0 (°/sec)	q_0 (°/sec)	r_0 (°/sec)
A	76.2	2.38	276.9	26.9	-5.6×10^{-5}	1.61	0.64	0.13	0.34
B	46.4	1.05	221.9	16.3	-1.3×10^{-2}	-1.68	-5.2×10^{-3}	-5.8×10^{-2}	-0.12
C	20.3	0.63	267.9	11.9	1.6×10^{-2}	0.0	1.0×10^{-2}	-0.43	5.9×10^{-2}

4.6.2 Simulation Run Matrix

Ten runs were made to demonstrate the characteristics of the DI controller. This set of simulations is summarized in Table 4.6. The test matrix consists of three subsets of test objectives. The first set, Runs 1 through 4, shows the vehicle responses during typical maneuvers performed in each portion of the flight envelope using the nominal set of aerodynamics coefficients. The second set, Runs 5 through 9, shows how uncertainties in aerodynamic coefficients affect the performance of the DI controller. Finally, Run 10 shows the effectiveness of the controller in coping with a side gust (i.e., acting essentially in the xy-plane of the vehicle), an outside disturbance for this mathematical model.

Table 4.6 Simulation Run Matrix

Run No.	Flight Cond.	Input	Remarks
1	A	-5° α-Step, 25° ϕ-Singlet	Nominal Case
2	B	-0.05°/sec α-Ramp, 7.5° ϕ-Singlet	Nominal Case
3	C	7.5° ϕ-Singlet	Nominal Case
4	A	7.5° ϕ-Singlet	Nominal Case with $K_\phi = 0.4$
5	A	10° α-Singlet, 10° ϕ-Singlet	Nominal Case
6	A	10° α-Singlet, 10° ϕ-Singlet	30% Aero Uncertainty*
7	A	10° α-Singlet, 10° ϕ-Singlet	50% Aero Uncertainty*
8	A	10° α-Singlet, 10° ϕ-Singlet	60% Aero Uncertainty*
9	A	10° α-Singlet, 10° ϕ-Singlet	60% Aero Uncertainty*, K_ϕ=0.4
10	C	Side Force Gust (50 ft/sec max)	Side Force Gust

*See Table 4.7 for complete definition.

Table 4.7 Aerodynamic Uncertainty Matrix

	DL_b	DL_p	DL_r	DL_{da}	DL_{dr}	DM_a	DM_q	DM_{de}	DN_b	DN_p	DN_r	DN_{da}	DN_{dr}
30%	+.3	-.3	+.3	-.3	+.3	-.3	+.3	-.3	+.3	-.3	+.3	-.3	+.3
50%	+.5	-.5	+.5	-.5	+.5	-.5	+.5	-.5	+.5	-.5	+.5	-.5	+.5
60%	+.6	-.6	+.6	-.6	+.6	-.6	+.6	-.6	+.6	-.6	+.6	-.6	+.6

4.6.3   Nominal Performance
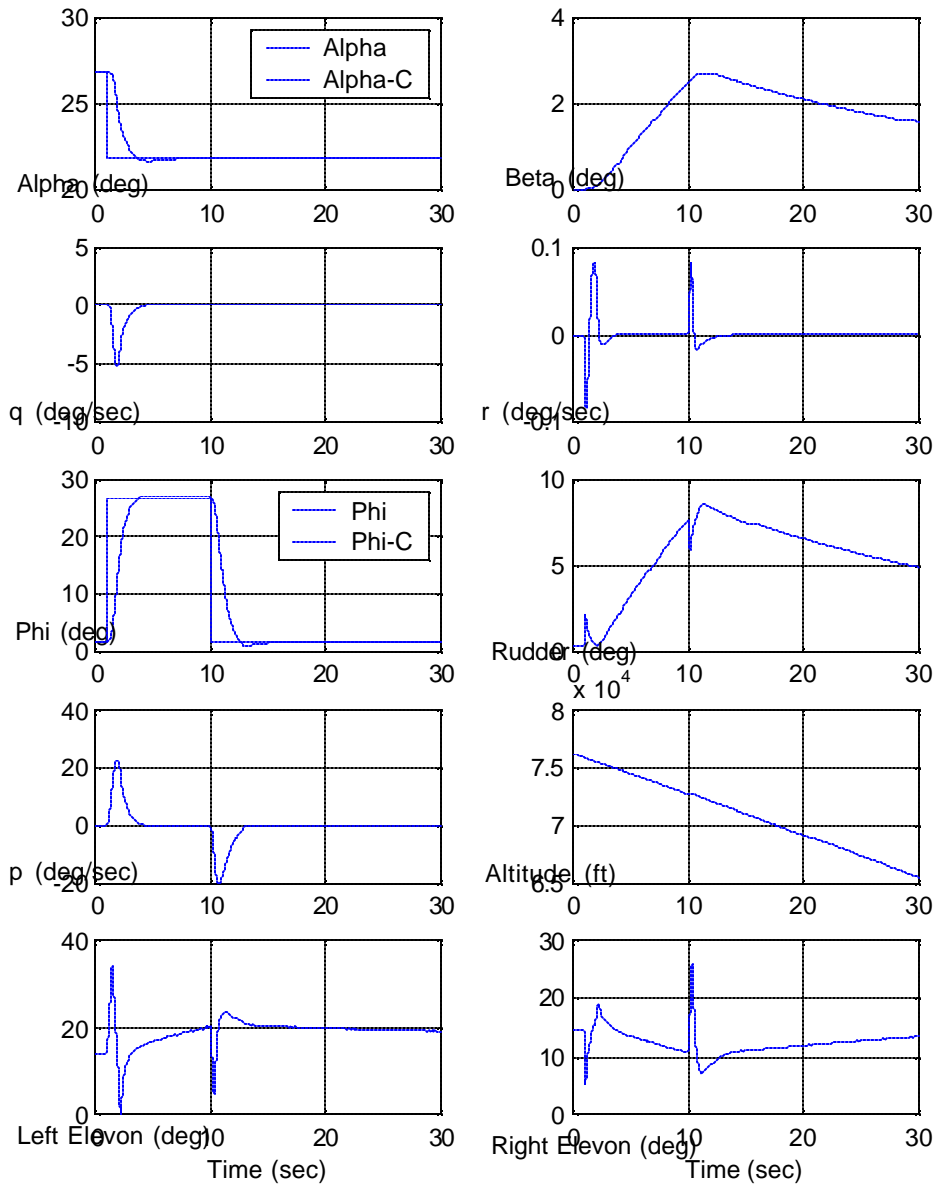
Figure 4.10 Simulation Run 1, supersonic flight ($M_\infty = 2.38$)

Run 1 (Figure 4.10) is the nominal case for supersonic flight conditions. A rather aggressive step input of –5° in α-command and a 25° singlet input in ϕ-command were used as command excitations. Both inputs were engaged at 1 second, and the duration of the singlet was set for 9 seconds. Despite the aggressive commands, the vehicle behaved very well both longitudinally and laterally/directionally, with little overshoot about any axis. Also, no actuator saturation occurred with either rudders or elevons throughout the simulation.

Figure 4.11 Simulation Run 2, transonic flight

Run 2 (Figure 4.11) demonstrates nominal transonic performance. This time a ramp of only – 0.05 deg/sec for α and a 7.5° singlet for ϕ were used as command inputs. Despite the fact that the

controller was unchanged from the previous case, the vehicle again responded very well and no actuators saturated.
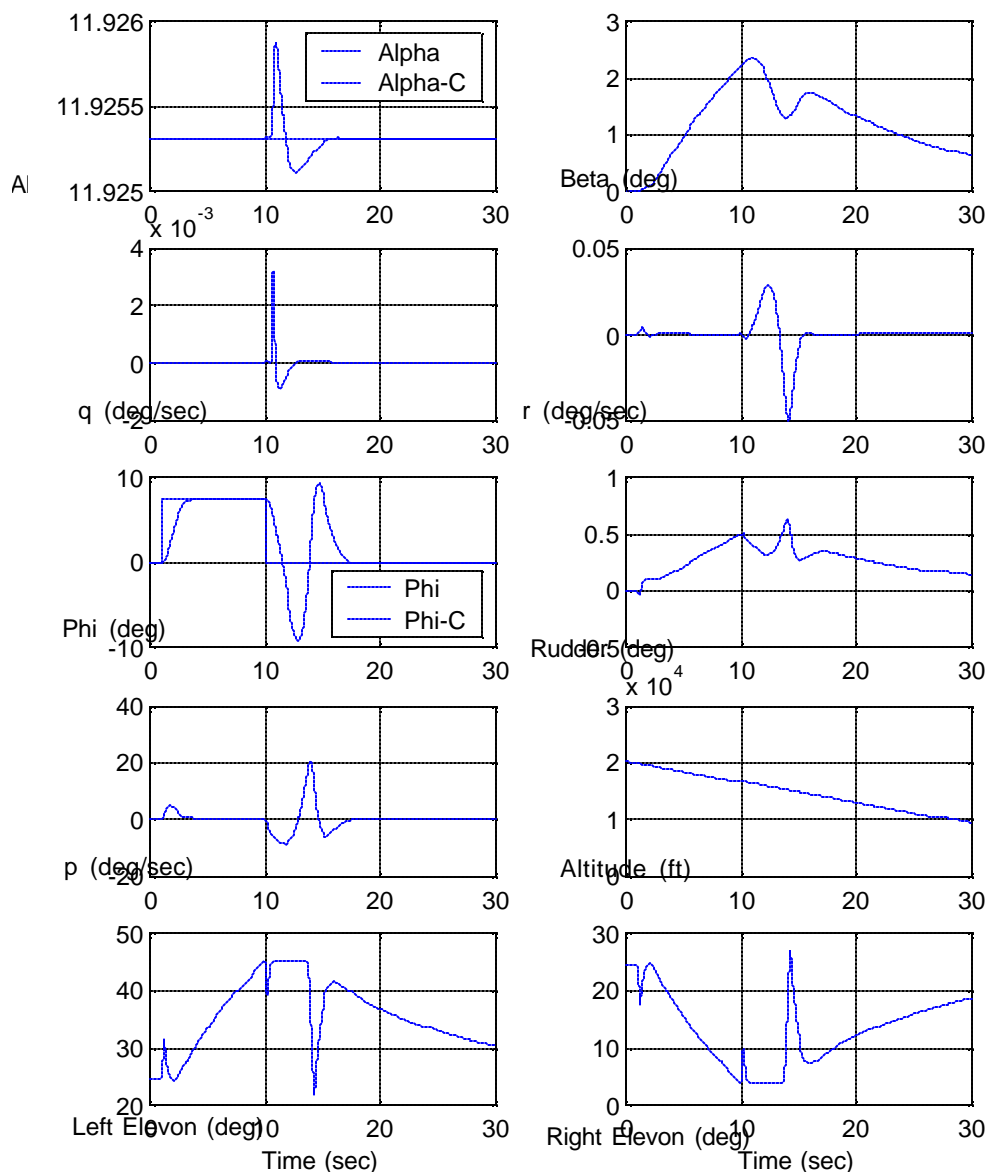
Figure 4.12 Simulation Run 3, subsonic flight, original unity outer loop gain
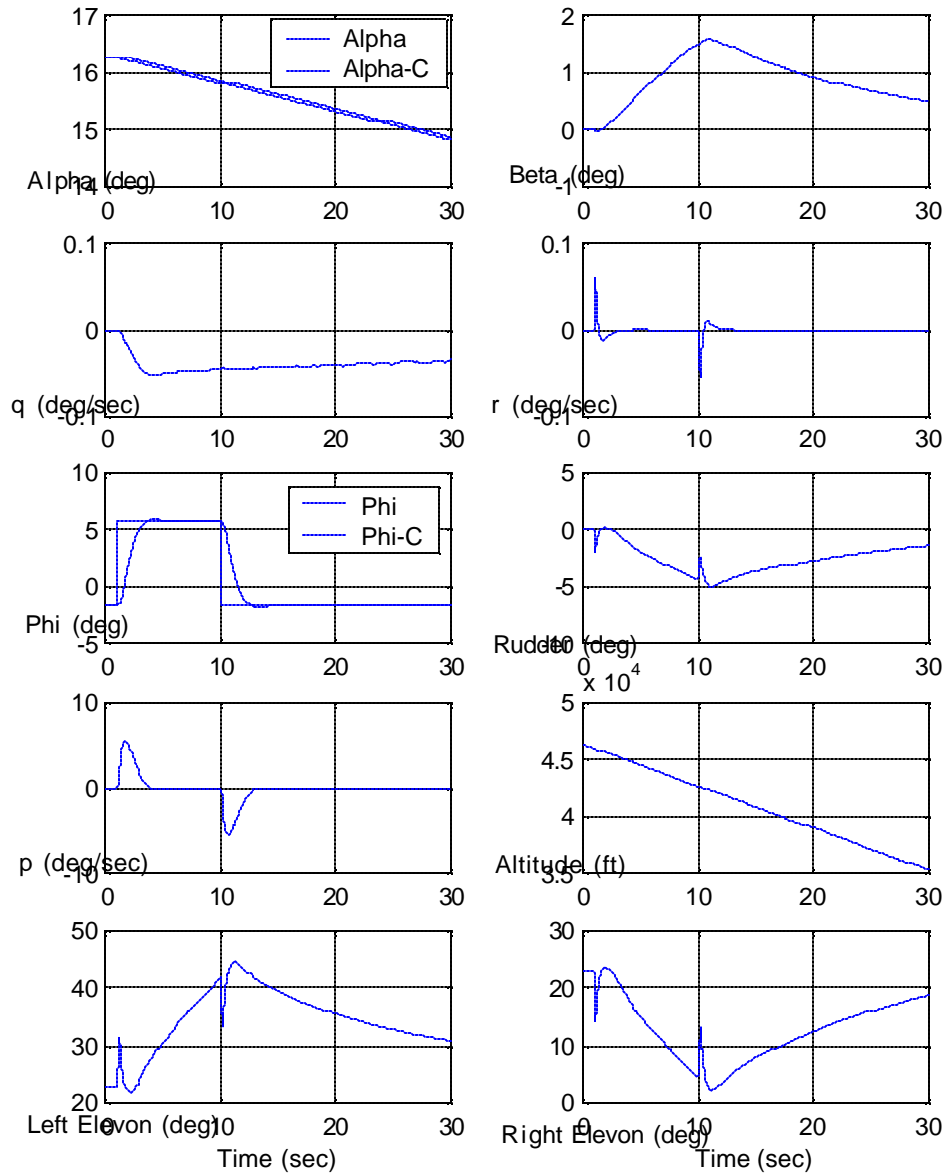
Run 3 (Figure 4.12) illustrates performance of the DI controller in subsonic flight. α was commanded to maintain its initial value while ϕ was commanded to follow a 7.5° singlet. The controller functioned well until about 10 seconds into the simulation, when elevon saturation occurred and the performance degraded rapidly especially in the roll axis. The roll axis degraded more than the pitch axis because, as explained previously, the control surface management logic puts priority on the pitch axis over the roll axis. There is little augmented damping in roll during the elevon saturation period. To overcome this deficiency, the gain of the bank angle outer loop was reduced from 1 to 0.4, and the

simulation was repeated using the same inputs. Output from this modified controller is shown in Figure 4.13 (Run 4). Although the elevons still saturate, the overall performance is improved over the unity feedback gain case.
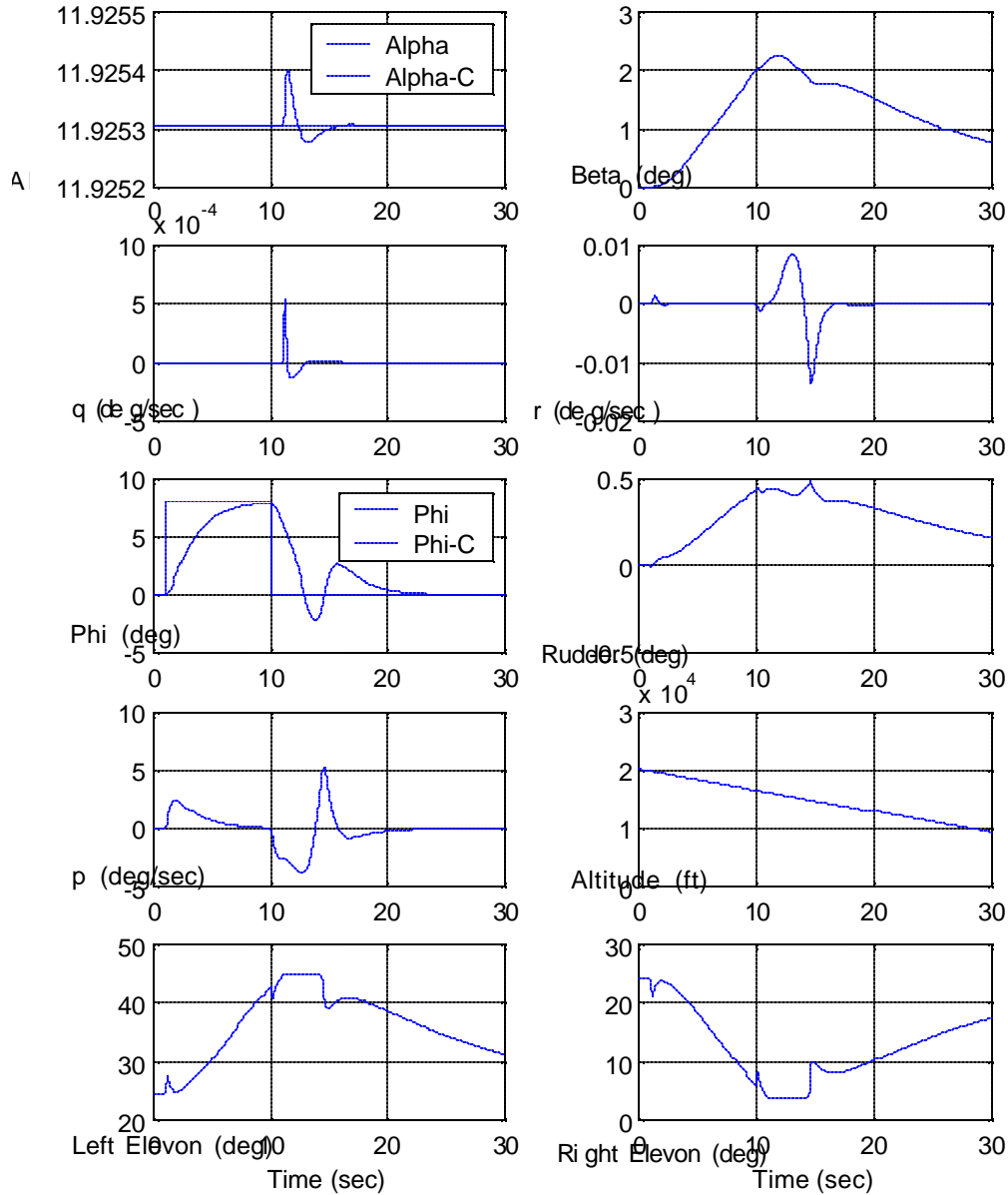
Figure 4.13 Simulation Run 4, subsonic flight, outer loop gain = 0.4

One point of emphasis: Although DI advocates usually assert no gain scheduling is needed because of DI's "global linearization" structure, in reality gain adjustments are still required whenever commands are large enough to saturate actuators. DI performs poorly during actuator saturation because only the aircraft dynamics are inverted and actuator dynamics are simply left out of the controller formulation (at least in this implementation). In other words, some optimization scheme that

recognizes actuator constraints is needed to overcome saturation problems. This drawback has more to do with the control surface allocation scheme than it does with the DI design itself. However, this result clearly suggests that the control allocation issue needs to be attacked by treating actuators as nonlinear elements and including them in the DI process. This effort will be handled in a future study.

4.6.4   Uncertainties in Aerodynamic Coefficients

The next issue to be discussed is uncertainty in aerodynamic coefficients. Out of the many parameters used in the simulation, 13 major aerodynamic coefficients were selected; they are presented in Table 4.5. The uncertainties are represented as algebraic additions to each parameter, and the magnitude of the uncertainties is set to three different percentages: 30, 50, and 60. The signs of these uncertainties, either an addition or a subtraction to/from the nominal values, are randomly chosen and are also shown in Table 4.7.

This set of simulations consists of five runs, one run for each nominal case as well as 30% and 50% uncertainty cases and two runs for the 60% case. The same set of flight conditions (A in Table 4.6) is used for all five runs. The same set of command inputs, 10° singlet commands for both α and ϕ, is also applied for each run.

The controller used for Runs 5 thorough 8 is the same controller that is used for Runs 1 thorough 3. The controller used for Run 4 has the same structure, but the gain of its outer ϕ-loop is reduced from 1 to 0.4. This modified controller also is used for Run 9.

Run 5 (Figure 4.14) is the nominal case. The vehicle responded well to the commanded inputs and no actuator saturation took place.

Run 6 (Figure 4.15) is the 30% uncertainty case. Despite the right elevon saturation at 2 seconds into the simulation, the vehicle still behaved well.

Run 7 (Figure 4.16) is the 50% uncertainty case. The divergence from the nominal case is obvious. Overshoots are large, especially during the period when the elevon saturation occurs. The bank angle excursion is also large because of pitch axis priority in the control surface management scheme. The lack of roll control authority under these conditions is underscored in this simulation.

Next, uncertainty is increased to 60% in Run 8 (Figure 4.17). The vehicle obviously diverges from the nominal trajectory, probably catastrophically.

From the 50% uncertainty case results, as well as by looking at the right elevon time history in this case, it is clear that elevon saturation, coupled with the priority for pitch axis control, causes loss of control. A quick (though not necessarily optimal) solution to this problem is, as in the previous case, reduction in outer ϕ-loop gain. This approach is examined in the next simulation run (Figure 4.18).
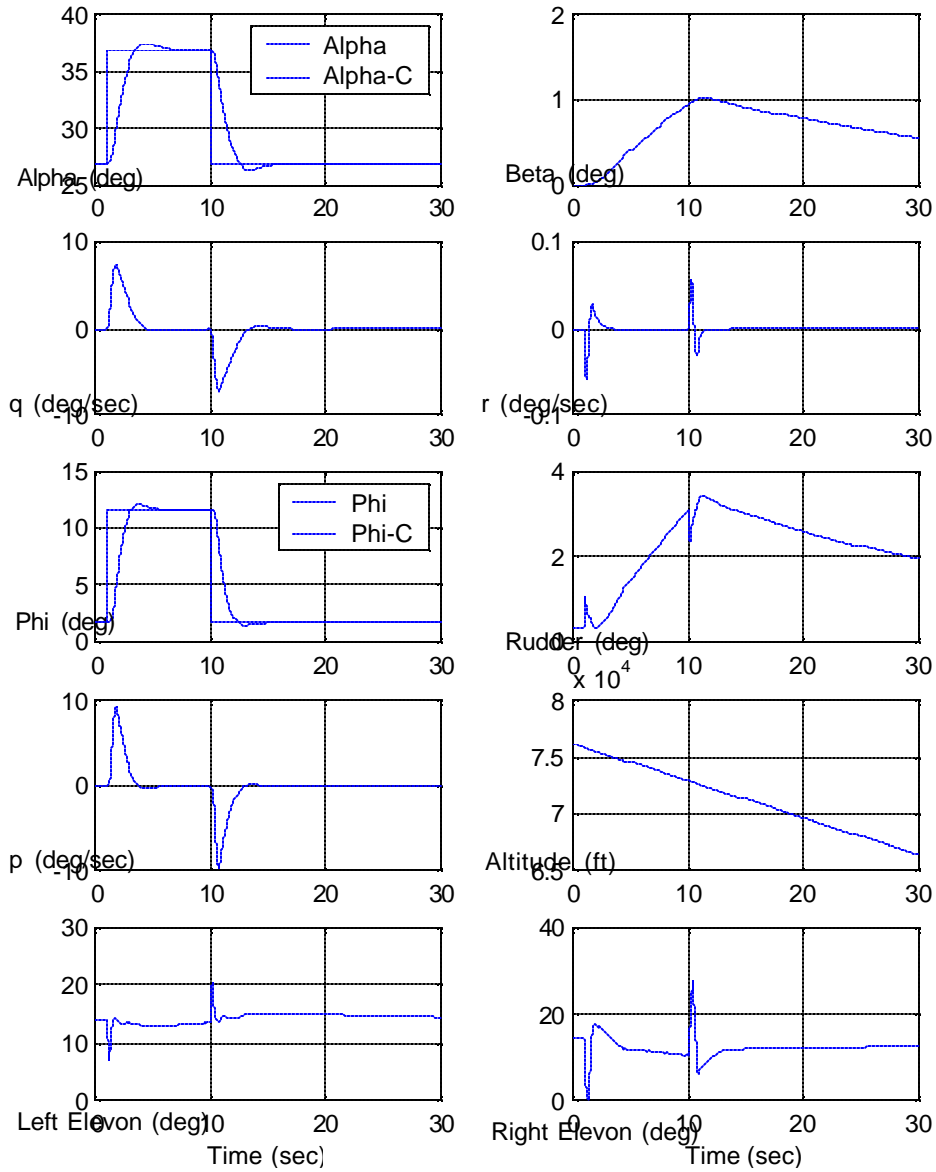
Figure 4.14 Simulation Run 5, supersonic flight (M_∞ = 2.38)

Figure 4.15 Simulation Run 6, supersonic flight ($M_¥$ = 2.38), 30% uncertainties
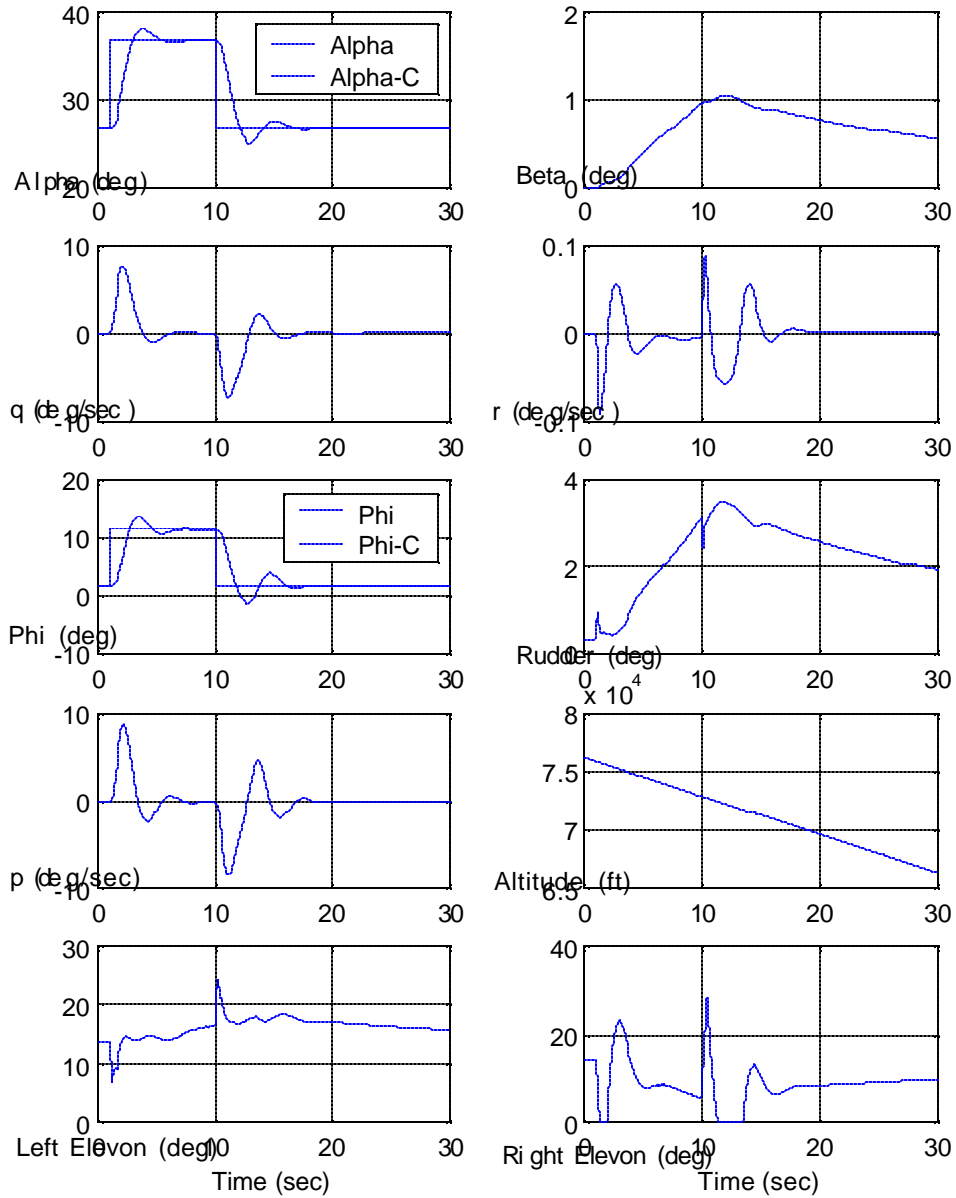
Figure 4.16 Simulation Run 7, supersonic flight ($M_{\Psi} = 2.38$), 50% uncertainties

By using the controller from Run 4 with the outer ϕ-loop gain reduced from 1.0 to 0.4, simulation Run 8 was repeated as Run 9. With this change in gain, the vehicle readily recovers from the upset, which suggests that gain modifications are effective in alleviating actuator saturation. Certainly the complexity of the controller increases as the number of gain alterations increases. There is, therefore, a tradeoff that designers have to consider between controller complexity and performance of the controller for the best controller design.
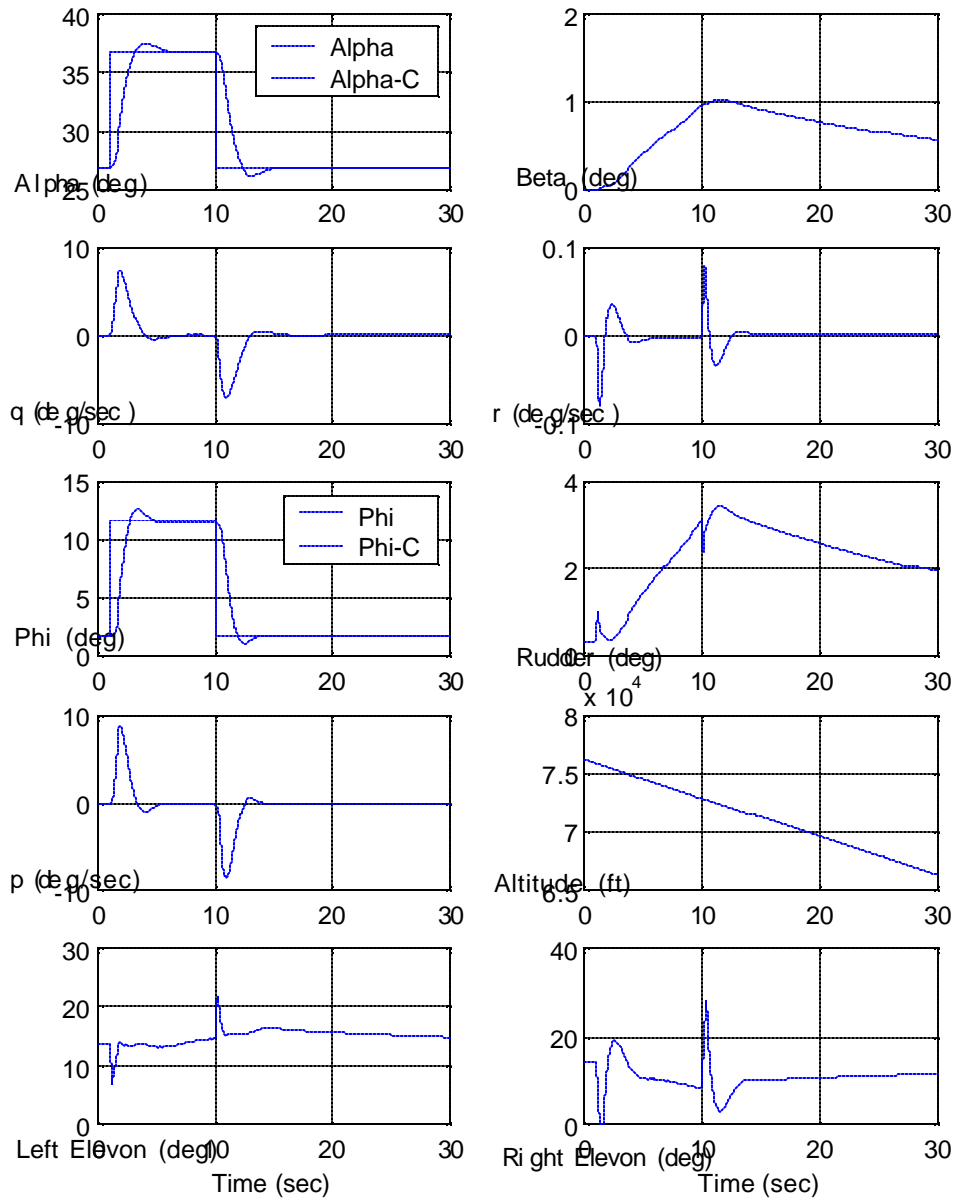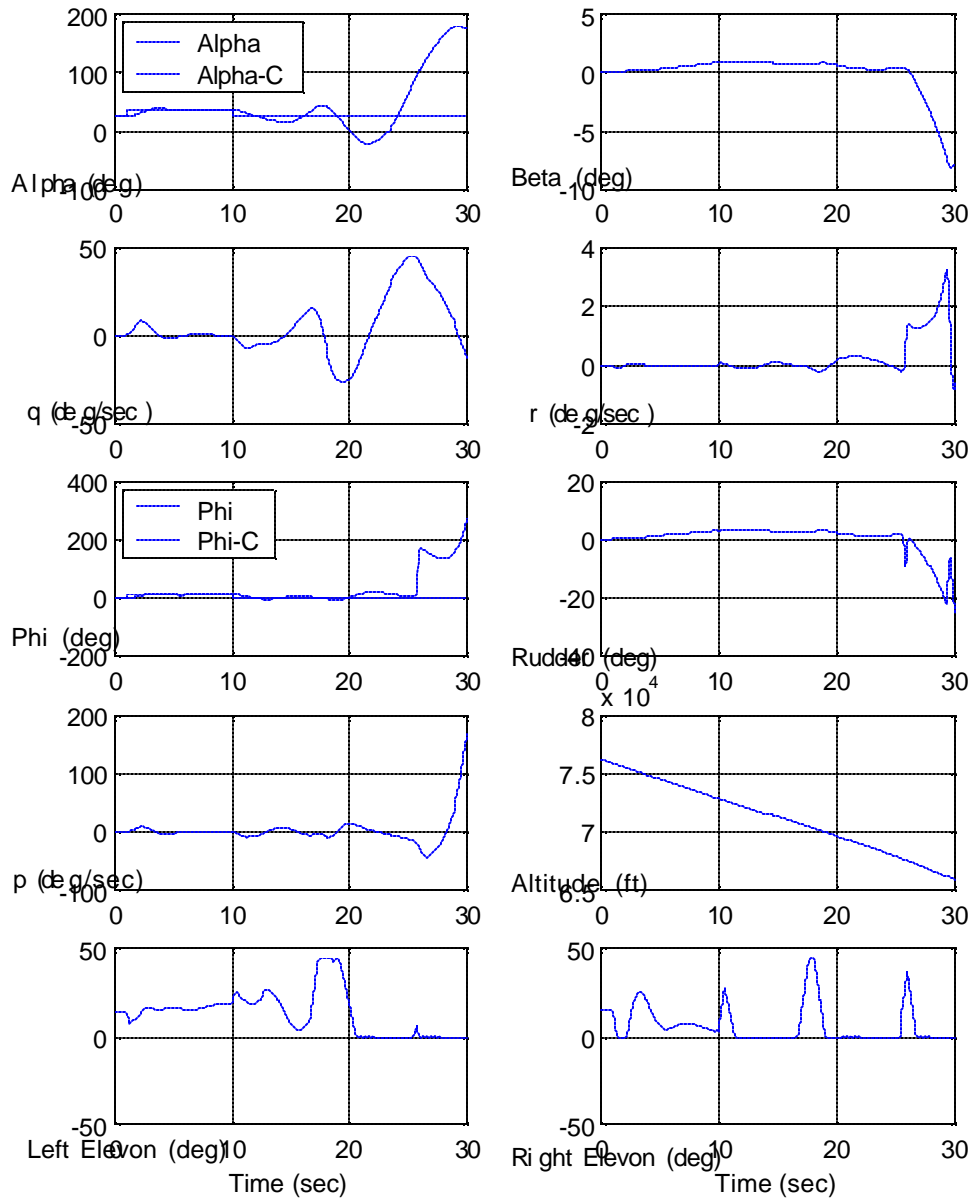
Figure 4.17 Simulation Run 8, supersonic flight (M∞ = 2.38), 60% uncertainties
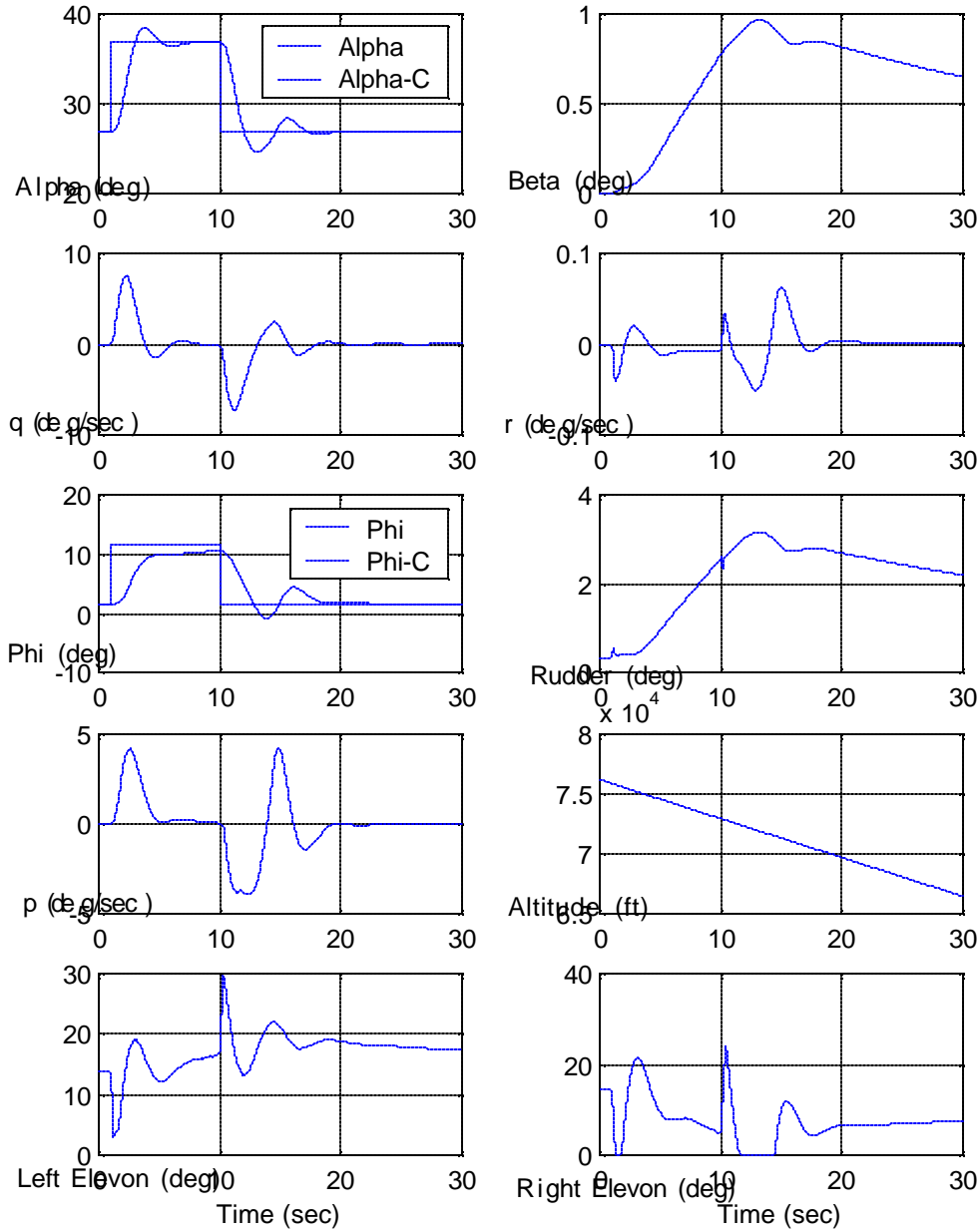
Figure 4.18 Simulation Run 9, supersonic flight (M_∞ = 2.38), 60% uncertainties, outer F-loop gain = 0.4

4.6.5   External Disturbances Effect: Side Gust

The final simulation example is an external disturbance case. We chose a side gust as a representative case of such disturbances and their effect on DI controllers.   The gust input shown in the upper left corner in Figure 4.19, Run 10, was added to the body y-axis velocity of the vehicle. The gust reaches up to 50 ft/sec, and the overall duration of the gust is 4 seconds. The flight condition used is the subsonic case (M_∞ = 0.63), Flight Condition C, shown in the Table 4.6.

The side gust disturbance input produces a maximum sideslip angle of about 1°. Also, bank angle is excited up to about 12° because of yaw-roll coupling. Bank angle did damp out rapidly (after approximately 1 cycle), and the sideslip angle became essentially negligible roughly 10 seconds after the gust was introduced. Overall, this DI controller worked well for rejecting external disturbances.

Figure 4.19 Simulation Run 10, subsonic flight ($M_{\Psi} = 0.63$),
external disturbance: side gust

51

4.7   Design Example 2

4.7.1   Introduction

Even in instances where the control input matrix $g(\mathbf{x})$ shown in Equation (3.4) is invertible, problems will arise if the matrix is very small in magnitude, indicating reduced control effectiveness on the state dynamics. Should this occur, the control becomes unbounded, causin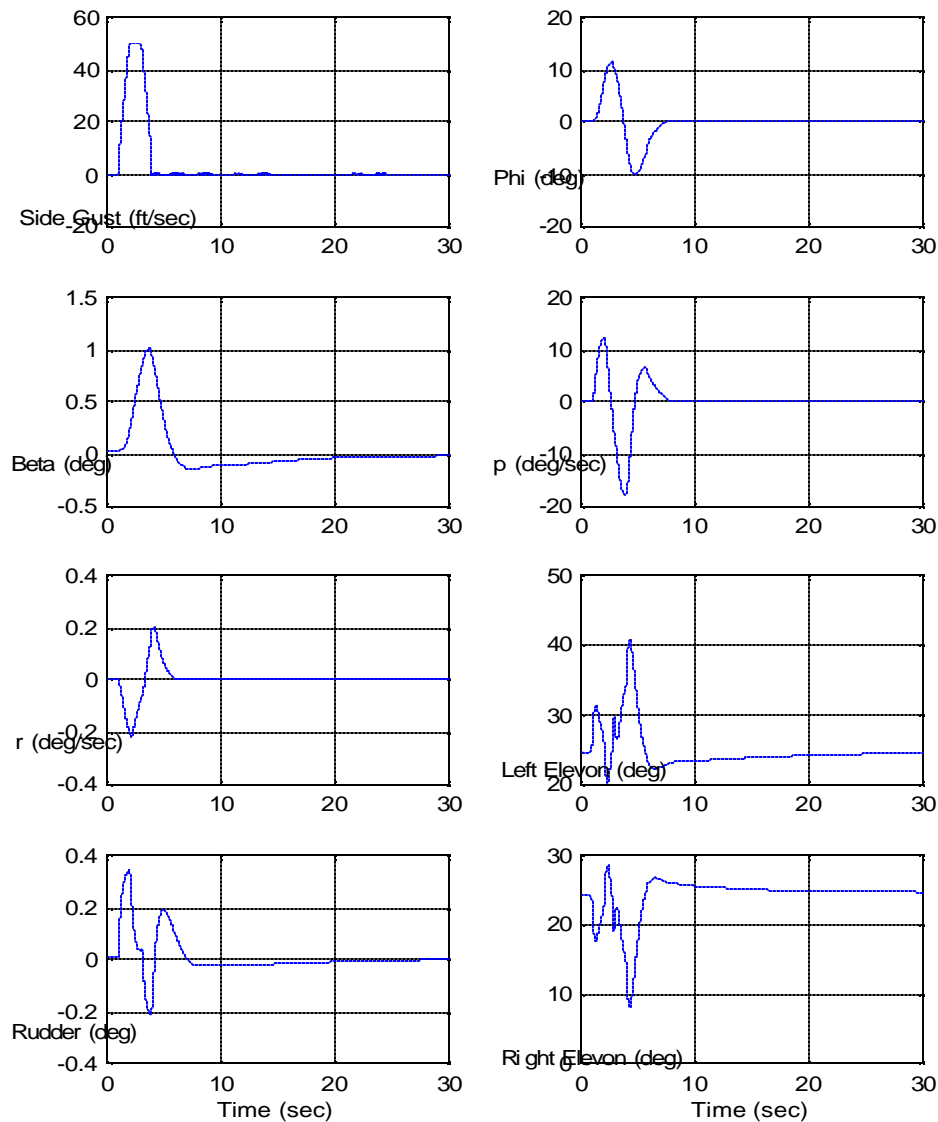g actuator saturation. For example, a control ineffectiveness problem is typically encountered in the inversion of angle-of-attack dynamics because the vehicle's longitudinal control surface has little effect on the angle-of-attack rate. A 2-time scale method has been developed and applied in previous research to bypass this problem.[17,18]   In our method, the control surface is used to generate the pitch rate dynamics, $\dot{q}$, directly. The resulting pitch rate is then used to control $\dot{\alpha}$. Figure 4.20 illustrates this approach in which two DI loops are present: a fast inner-loop inversion for rotational rate variables, such as q, and a slow outer-loop inversion for rotational variables, such as α.
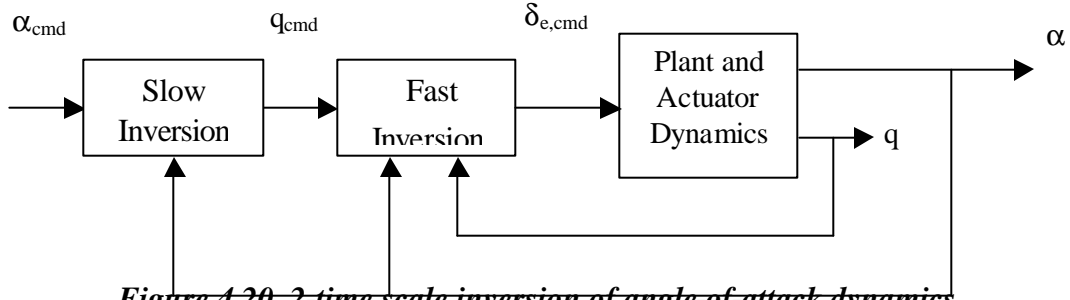
Figure 4.20 2-time scale inversion of angle-of-attack dynamics

Equation (4.16) shows the pitch rate command generated by the slow outer-loop inversion, and Equation (4.17) shows the elevon command generated by the fast inner-loop inversion, required to produce the desired angle-of-attack response. It should be noted that both control laws require angle-of-attack and pitch rate feedback, which is full-state feedback for the short period approximation.

$$q_{cmd} = \left(1 + \frac{Z_q}{U_o}\right)^{-1}\left(\dot{\alpha}_{des} - \frac{Z_\alpha}{U_o}\alpha\right) \qquad (4.16)$$

$$\delta_{e,cmd} = M_{\delta_e}^{-1}\left(\dot{q}_{des} - M_\alpha\alpha - M_q q\right) \qquad (4.17)$$

The control laws shown above are used to invert the inherent pitch rate dynamics and the angle-of-attack dynamics, respectively. The proposed forms of desired dynamics developed in Section 3.6 are applied separately to each of the short-period state dynamics before they are evaluated in terms of

- Time domain performance,
- Stability robustness and performance robustness,
- Effect on motion sickness,
- Quadratic cost function, and
- Passenger ride comfort index.

4.7.2   Time Domain Design Requirements

The time domain performance specifications used to evaluate vehicle response are short-period damping ratio and natural frequency. The requirements for the X-38 vehicle are selected from Mil-STD-1797A[14] to satisfy Level 1 flying qualities for a Class II vehicle during a Category B flight phase. These Level 1 standards bound the short-period damping ratio and natural frequency as follows:

$$0.30 \leq \zeta \leq 2.0 \tag{4.18}$$

$$0.7 \mathrm{s}^{-1} \leq \omega_n \leq 1.3 \mathrm{s}^{-1} \tag{4.19}$$

The time domain response requirements for a step input are shown pictorially in Figure 4.21. These requirements satisfy Level 1 flying qualities for the vehicle.

Command	Amplitude			
	K1	K2	K3	K4
Alpha to 2.0°	0.63	0.9	1.1	1.2
Roll to 10° for Mach < 1.0	0.63	0.9	1.1	1.2
Roll to 10° for Mach = 1.0	0.63	0.9	1.1	1.2

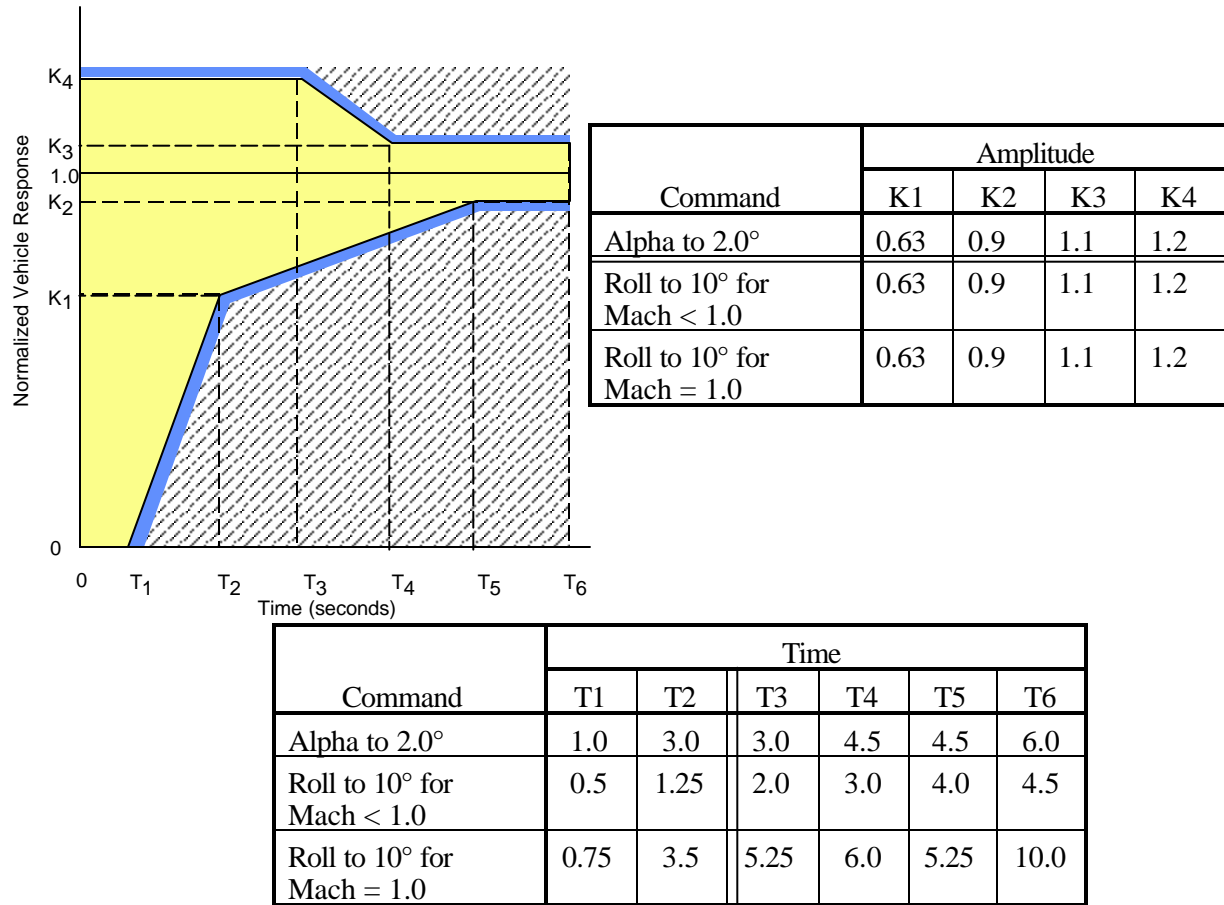| Command	Time					
	T1	T2	T3	T4	T5	T6
Alpha to 2.0°	1.0	3.0	3.0	4.5	4.5	6.0
Roll to 10° for Mach < 1.0	0.5	1.25	2.0	3.0	4.0	4.5
Roll to 10° for Mach = 1.0	0.75	3.5	5.25	6.0	5.25	10.0

Figure 4.21 Time domain performance specifications

Two numeric indices are also used to evaluate the responses of each controller. A quadratic cost function, J, (shown in Equation (4.20)) is used to evaluate the magnitude of each controller response in terms of the four longitudinal states and the elevon input.

$$J = \mathbf{x}^T Q \mathbf{x} + \mathbf{u}^T R \mathbf{u} \qquad (4.20)$$

It was desired to weight angle-of-attack more heavily than pitch rate because angle-of-attack is the variable being controlled. Since the behavior of the phugoid mode is less important, the velocity and pitch attitude states are weighted less than the short period states. Therefore, the weighting matrices used are: $Q = \text{diag}(.1 \quad 10 \quad 1 \quad .1)$ and $R = 1$. A passenger ride comfort index[19], shown in Equation (4.21), was also used to evaluate the responses.

$$C = 2.1 + 17.2\bar{a}_z \qquad (4.21)$$

The RMS vertical acceleration, $\bar{a}_z$, is $a_z = U_0\dot{\alpha} - U_0 \cos(\alpha_0)q + g\sin(\theta_0)\theta$ in which vertical acceleration subscripts denote trim values. Lower comfort indices indicate increased passenger ride comfort.

4.7.3   Controller Design

The proportional, PI, and flying qualities dynamics desired are selected to satisfy Equations (4.18) and (4.19). The ride qualities dynamics were selected to satisfy the desired CAP and damping ratio values shown in Figure 3.13. Table 4.8 summarizes the desired dynamics selected for these controllers. Each set of dynamics acts on the error between CV command and its feedback term. The robust outer loop was designed via pole placement through the use of MATLAB's *place* command. This outer loop consists of full-state feedback that is operated on by a matrix of gain values. The poles were placed according to the desired closed-loop pole locations.

Table 4.8 Desired Dynamics Selection

Desired Dynamics	Angle-of-Attack Case	
	Slow inversion	Fast Inversion
Proportional	0.8	1.3
Proportional Integral	$6\left(\frac{1}{2}CV_{cmd} - CV\right)$ $+\frac{6.25}{s}(CV_{cmd} - CV)$	$\left(\frac{1}{2}CV_{cmd} - CV\right)$ $+\frac{2}{s}(CV_{cmd} - CV)$
Flying Qualities	$\dfrac{1.4(s+0.8)}{s^2 + 2.24s + 1.96}$	$\dfrac{1.5(s+0.8)}{s^2 + 2.24s + 1.96}$
Ride Qualities	$\dfrac{1.96}{s+3}$	$\dfrac{7}{s+4}$

4.7.4   Time Domain Analysis

Time histories for the inverted angle-of-attack dynamics are shown in Figure 4.22. The PI dynamics respond very fast, causing a large actuator rate initially. Although the flying qualities dynamics slightly violate the angle-of-attack time domain constraints, the other forms of dynamics stay within the time-domain specifications. It should be noted that all responses satisfy the actuator position and rate constraints. Of the various desired dynamics, the proportional form results in the lowest cost and lowest comfort index for an angle-of-attack DI controller. The resulting costs and comfort indices of these responses are shown in Table 4.9.

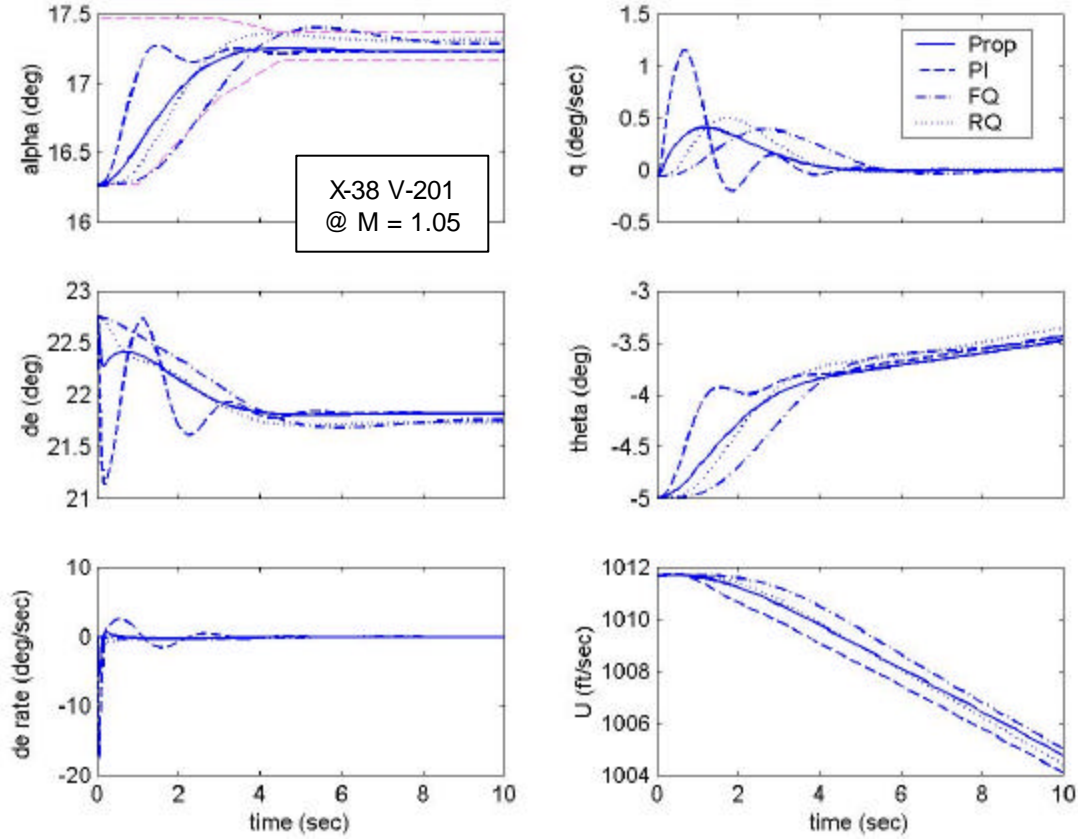*Figure 4.22 Time histories for the inverted* **a** *dynamics*

Table 4.9 Cost and Passenger Comfort Index

Desired Dynamics	Cost	Comfort Index
Proportional	130.4	2.56
Proportional Integral	172.0	2.60
Flying Qualities	142.4	2.58
Ride Qualities	139.7	2.61

4.7.5   Frequency Domain Analysis

Robustness is a key element in the reentry vehicle flight control system design because of the broad flight envelope in which these vehicles must operate. The robustness technique that is used to analyze controllers developed in this work is adapted from current research in robustness methods for the X-38 vehicle[20]. A sigma-Bode plot of the loop gain singular values is used to evaluate robustness over a range of input frequencies. Further explanation of the sigma-Bode plot can be found in Section 4.8.2. The performance and robustness criteria used to evaluate the controllers in this section are

- Zero steady-state error.
- Attenuation of low-frequency disturbances by a factor of 0.1.
- Linear model accuracy to within 10% of actual plant for frequencies up to 2 rad/sec and growth without bound at 20 dB/decade thereafter.

To satisfy these requirements, illustrated in Figure 4.23, the singular values of the loop gain must lie outside of the performance requirement and stability requirement areas for all frequencies.
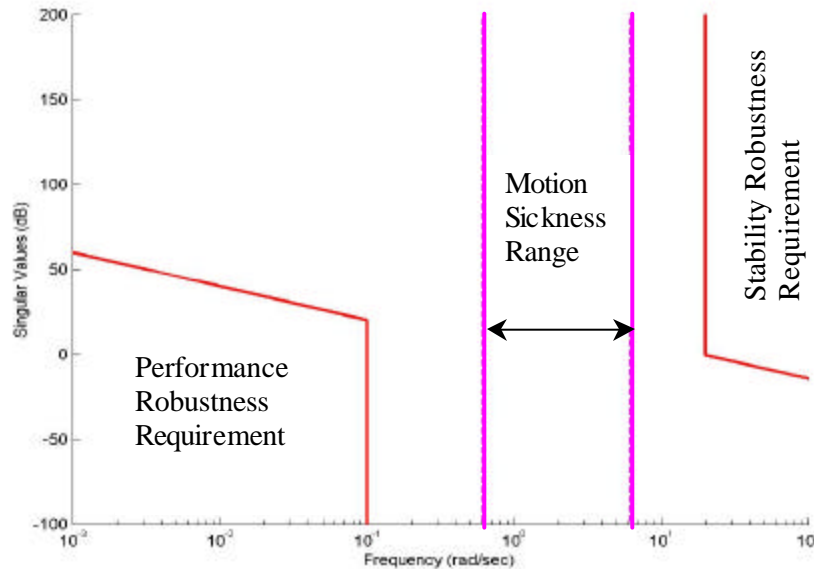

Figure 4.23 Robustness constraints

An additional requirement on signal attenuation can be added to the singular value plot of Figure 4.23. For this example, attenuate the signals between 0.6 rad/sec to 1.6 rad/sec in order to alleviate the passenger motion sickness that can occur at this frequency range[19]. Since the singular values of a matrix represent the relative size of a mode, attenuate the singular value response within this frequency range. The singular value response within this range should be continuously decreasing. Any amplification within it would increase the passengers' motion sickness.

The sigma-Bode plots for these cases are shown in Figure 4.24. Both the proportional and the PI desired dynamics violate the stability robustness requirement above 20 rad/sec. These dynamics also violate the attenuation requirement toward the higher end of the frequency range. Figure 4.24 clearly indicates that flying qualities compensation and ride quality compensation are sufficient for a robust angle-of-attack DI controller because they satisfy stability robustness, performance robustness, and the

56

motion sickness constraint. Table 4.10 summarizes the results of each set of desired dynamics in terms of satisfying time domain and frequency domain requirements.
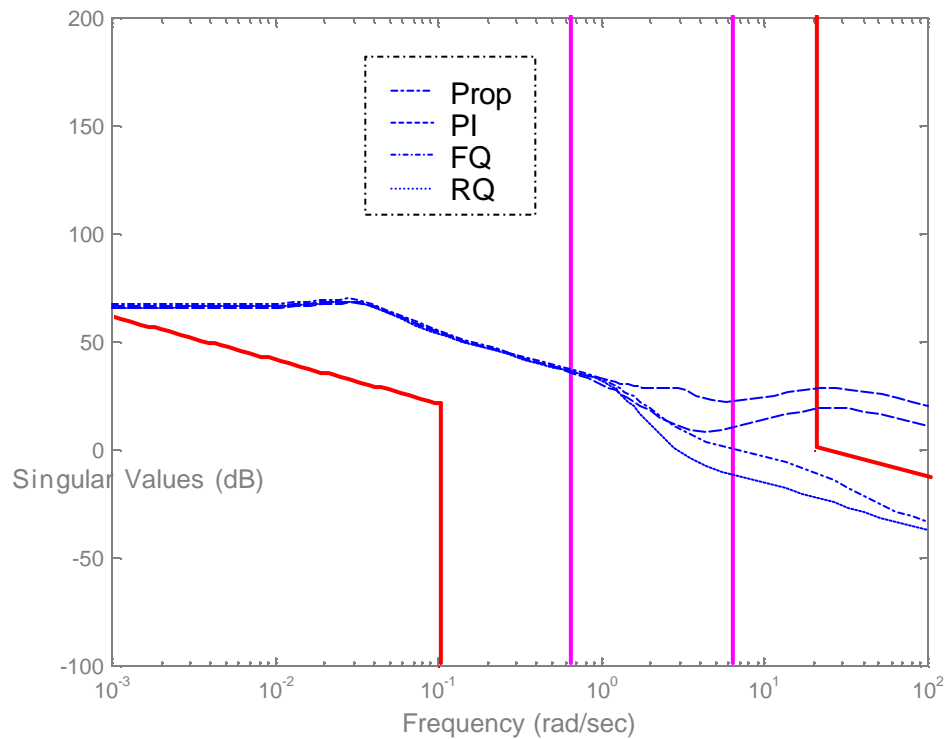
Figure 4.24 Sigma-Bode of closed-loop system

Table 4.10 Summary of Compliance with Design Specifications

Desired Dynamics	Step Response	Control Responses	Robustness Constraints	Motion Sickness
Proportional	3	3	5	5
PI	3	3	5	5
Flying Qualities	5	3	3	3
Ride Qualities	3	3	3	3

4.8   Design Example 3

4.8.1   Introduction

A systematic way to synthesize and analyze the robustness of a DI-based controller in a straightforward manner must be found so that engineering intuition can be easily applied throughout

57

the design process. The approach presented in this study uses LQG techniques[21] to synthesize the outer-loop controller. This approach is a relatively easy-to-use design method for multivariable control design. It is a loop-shaping tool in the frequency domain that gives robustness to the system[22-25]. Desired dynamics are given by a dynamic compensator that shapes the loop. Selected dynamics are based on performance and stability robustness requirements. These requirements are straightforwardly formulated during synthesis of the controller as frequency-dependent singular value bounds (Figure 4.25).
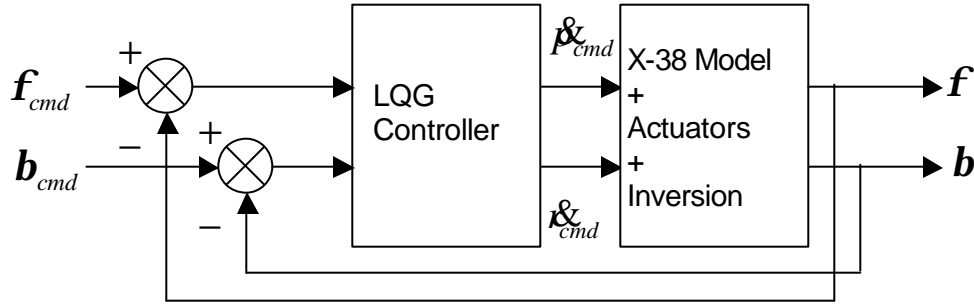

Figure 4.25 X-38 lateral-directional control system

4.8.2   Design Requirements

The controller design procedure is iterative and centers on designing a controller that will satisfy a set of specifications. Specifications are usually stated in both time and frequency domains. A time domain requirement for the X-38 lateral-directional system is defined by the time histories of a 10° bank angle response after a step input[26]. The boundaries of this requirement are depicted in Figure 4.21. Two different Mach-number-dependent requirements are specified in the time domain. Clearly, requirements for the subsonic regime are much tighter than for supersonic flight conditions. According to the requirements[26], no sideslip angle constraint is spelled out for lateral-directional vehicle control. Nevertheless, there should be as small a sideslip perturbation as possible to maintain a coordinated turn at all times. The DI inner-loop controller cancels the existing system dynamics and replaces them with designer-specified responses. Since desired outputs are often decoupled about each axis, off-diagonal coupling effects are typically minimal for a good DI controller.

Next, frequency domain specifications are meant to ensure performance and stability robustness. Both of these requirements are expressed using singular values. Singular value is a suitable choice to express the magnitude of matrix functions because it generalizes known SISO statements and constraints of the design problem to MIMO cases. In general, a singular value is thought of as the Bode magnitude plot for an SISO case, but singular values extend the concept to a MIMO system as well. Therefore, MIMO design is carried out using classical control design concepts that align with engineering intuition.

To achieve an adequate response, the gain in the low-frequency region must first be high enough to give a quick response to the input while the slope of the singular values must be steeper than –20 dB/decade to reduce the steady-state error. Further, we assume it is desirable to have at least 0.1 rad/sec of crossover frequency to obtain a good closed-loop transient response. These requirements are integrated to form the singular value bounds in the low-frequency region. This low-frequency "trapezoid" is sketched in Figure 4.26.
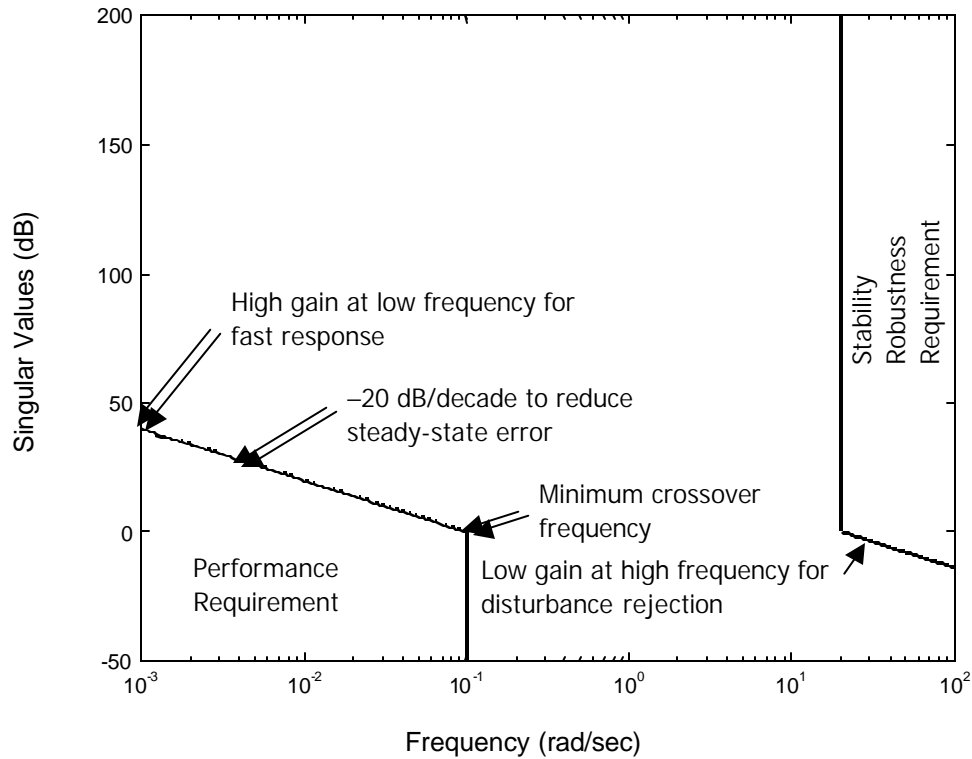
Figure 4.26 Frequency domain requirements

At high frequencies, singular values are bounded by the unmodeled dynamics associated with high-frequency models such as flexible and vibrational models. These high-frequency models are often neglected when the plant model is being developed; the vehicle is often treated as a rigid body. As a result, there is a difference between the assumed mathematical plant, $G(s)$, and the actual plant, $G'(s)$. This difference is defined and described as model uncertainty[27]. Here, an unmodeled dynamics model, suggested by Stevens and Lewis[27], is used. This uncertainty model assumes the rigid body model is accurate to within 10% and up to a frequency of 2 rad/sec, after which the uncertainty grows at a rate of 20 dB/decade. This uncertainty model is expressed in the following transfer function and is used to model the uncertainties of the X-38 vehicle at high frequencies:

$$m(w) = \frac{s+2}{20} \tag{4.22}$$

We assume $m(w)$ to be bounded with uncertainty in the X-38 vehicle transfer function. Multiplicative uncertainty is expressed in terms of an assumed plant model and the actual plant by

$$G'(jw) = [I + M(jw)]G(jw), \tag{4.23}$$

where the unknown discrepancy satisfies a known bound.

59

$$\overline{s}\mathbf{C}_M\mathbf{b}_{jw}\mathbf{g} + \overline{\mathbf{t}}_{m}\mathbf{b}_{\mathbf{w}}\mathbf{g} \tag{4.24}$$

Under this assumption, for stability robustness with modeling errors, the loop gain referred to the output should satisfy:

$$\overline{s}\mathbf{C}_{GK}\mathbf{b}_{jw}\mathbf{g} + \frac{1}{m\mathbf{b}_{\mathbf{w}}\mathbf{g}} \geq \left|\frac{20}{s+2}\right| \tag{4.25}$$

when $1/m\mathbf{b}_{\mathbf{w}}\mathbf{g} < 1$. Here, K represents the compensator. This uncertainty bound, which is obtained from the unmodeled dynamics, is applied in the high-frequency region. Finally, the complete frequency domain bounds are shown in Figure 4.26 (previous page).

4.8.3   Lateral-Directional Dynamic Inversion Controller

Lateral-directional DI control equations were developed previously in Section 3.2.2. The following form, which is provided in Equation (3.12), is used for this example:

$$\begin{bmatrix} da \\ dr \end{bmatrix}^{cmd} = \begin{bmatrix} L_{da} & L_{dr} \\ N_{da} & N_{dr} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \dot{p} \\ \dot{r} \end{bmatrix}^{des} - \begin{bmatrix} L_p & L_r & L_b \\ N_p & N_r & N_b \end{bmatrix} \begin{bmatrix} p \\ r \\ b \end{bmatrix}^{meas} \right\} \tag{4.26}$$

4.8.4   Dynamic Inversion Inner-Loop Controller

The lateral-directional DI controller, together with the X-38 model and control surface blocks, forms the DI inner-loop augmented system. This augmented inner loop is

$$\dot{\mathbf{x}}_{IL} = \mathbf{A}_{IL}\mathbf{x}_{IL} + \mathbf{B}_{IL}\mathbf{u}_{IL}, \tag{4.27}$$

where the inner-loop state vector, $\mathbf{x}_{IL}$, is

$$\mathbf{x}_{IL} = \begin{bmatrix} b & p & r & f & \dot{da} & da & \dot{dr} & dr \end{bmatrix}^T \tag{4.28}$$

and the inner-loop control vector, $\mathbf{u}_{IL}$, is

$$\mathbf{u}_{IL} = \begin{bmatrix} \dot{p}^{des} \\ \dot{r}^{des} \end{bmatrix}. \tag{4.29}$$

The state matrix, $\mathbf{A}_{IL}$, and the control distribution matrix, $\mathbf{B}_{IL}$ for transonic flight (Table 4.5, Case B) are expressed numerically as follows:

$$
\mathbf{A}_{IL} =
\begin{bmatrix}
-7.8e-2 & 0 & -1.0 & 4.9e-2 & 0 & 0 & 0 & 1.5e-2 \\
-3.1e+1 & -2.3e-1 & 2.5e-1 & 0 & 0 & 3.1 & 0 & 9.3 \\
4.0e-1 & 2.2e-2 & -9.8e-2 & 0 & 0 & -8.5e-3 & 0 & -1.5 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
6.3e+3 & 1.9e+1 & 8.3e+1 & 0 & -3.7e+1 & -6.8e+2 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1.5e+2 & 1.0e+1 & -4.6e+1 & 0 & 0 & 0 & -3.7e+1 & -6.8e+2 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
$$

$$
\mathbf{B}_{IL} =
\begin{bmatrix}
0 & 0 & 0 & 0 & 2.2e+2 & 0 & -1.3 & 0 \\
0 & 0 & 0 & 0 & 1.4e+3 & 0 & -4.7e+2 & 0
\end{bmatrix}^{T}
$$

We assume measurements are limited to bank angle, ϕ, and to sideslip angle, β. Therefore, the following output equation results are:

$$
\mathbf{y} = \begin{bmatrix} f \\ b \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}_{IL} \tag{4.30}
$$

The block diagram representation of this DI inner-loop is shown in Figure 4.27. In addition, the singular values of the inner-loop versus frequency are plotted in Figure 4.28.
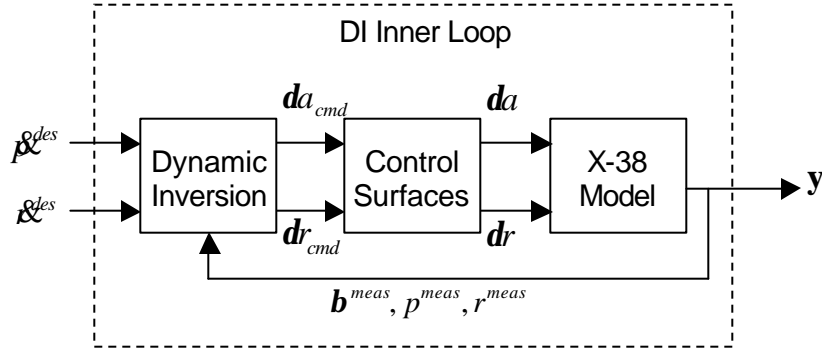

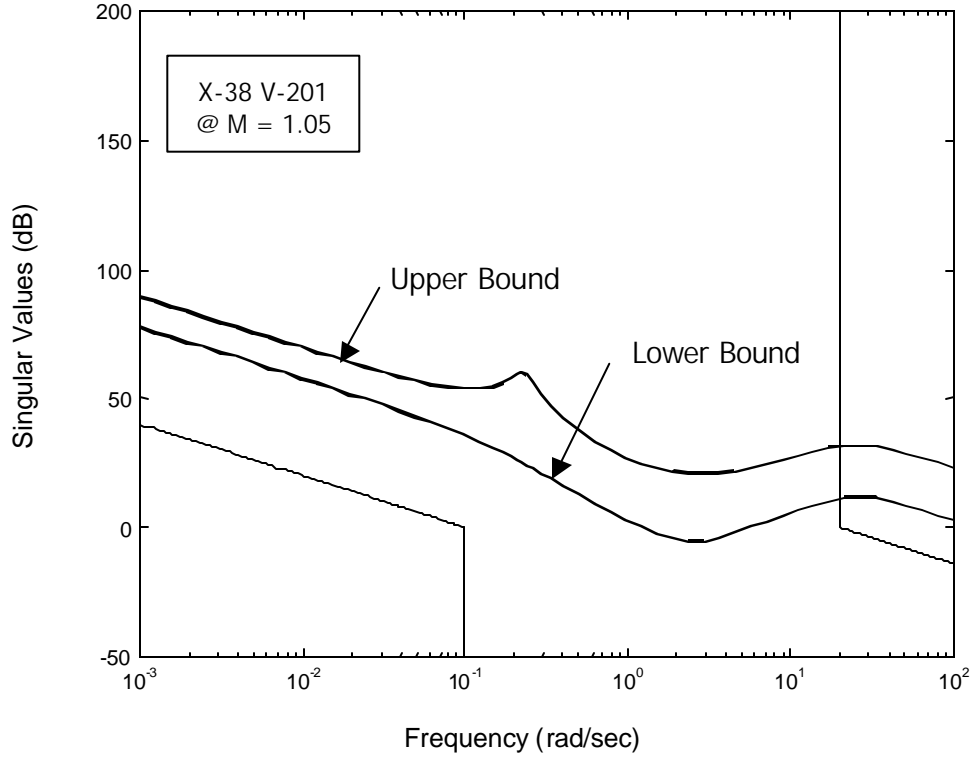*Figure 4.27 Dynamic Inversion Control Inner-Loop block diagram*

Figure 4.28 Singular values of the dynamic inversion inner-loop system

4.8.5   Augmented System

Since DI alone does not achieve the desired specifications, stability robustness requirements at high frequencies are not met. Integrators are added to each control channel to correct this deficiency. The X-38 plant, actuators, and DI controller are augmented to form the following system:

$$\dot{\mathbf{x}}_{aug} = \mathbf{A}_{aug}\mathbf{x}_{aug} + \mathbf{B}_{aug}\mathbf{u}_{aug} \tag{4.31}$$

where

$$\mathbf{x}_{aug} = \begin{bmatrix} \boldsymbol{b} & p & r & \boldsymbol{f} & \dot{d\!a} & d\!a & \dot{d\!r} & d\!r & \boldsymbol{e}_f & \boldsymbol{e}_b \end{bmatrix}^T \tag{4.32}$$

and

$$\mathbf{u}_{aug} = \begin{bmatrix} \dot{p}^{des} \\ \dot{r}^{des} \end{bmatrix} \tag{4.33}$$

62

In Equation (4.32), e_f and e_b are augmented states because of the addition of the integrators in the bank angle and the sideslip channels, respectively. The augmented matrices, $\mathbf{A}_{aug}$ and $\mathbf{B}_{aug}$, are obtained by performing the following manipulations:

$$\mathbf{A}_{aug} = \begin{bmatrix} \mathbf{A}_{IL} & \mathbf{B}_{IL} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \text{and} \quad \mathbf{B}_{aug} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_2 \end{bmatrix} \tag{4.34}$$

The corresponding singular value plot for this augmented system is shown in Figure 4.29. It suggests that the nominal design has now been reshaped to meet the specified frequency domain requirements. But we still have not tailored the outer loop for robustness, and we have not dealt with the LQR-LQG observer issue.
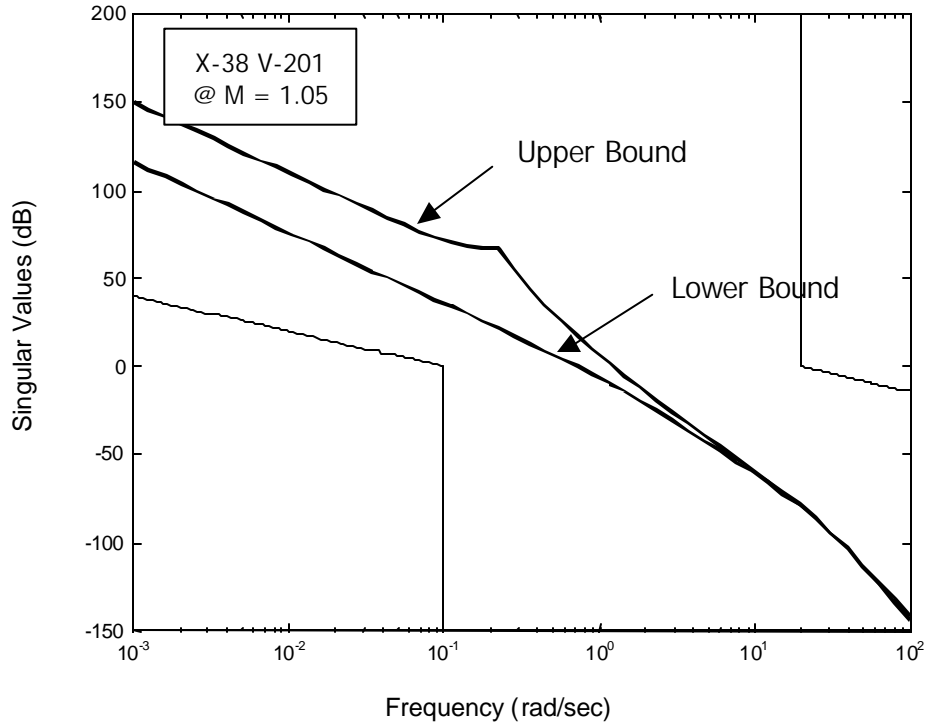

Figure 4.29 Augmented system singular values

4.8.6   Observer Design

The objective of this step is to create a fast dynamics observer that can be used with a regulator to form an LQG controller that will satisfy both performance and stability robustness requirements. The form of the observer is a Kalman filter designed for the augmented system of the previous section. The following weighting matrices were selected, by trial and error, so that the singular values of the resulting Kalman filter open-loop gain, $\mathbf{CF}(\mathbf{s})\mathbf{L}$, satisfy the singular value frequency domain requirements:

$$\mathbf{Q} = diag\left(1, \quad 1, \quad 1, \quad 1, \quad 1, \quad 1, \quad 1, \quad 1, \quad 1\times10^8, \quad 1\times10^8\right) \text{ and } \mathbf{R} = 0.005\mathbf{I}_2$$

where $\mathbf{F}(\mathbf{s}) = (s\mathbf{I} - \mathbf{A})^{-1}$.

The resulting Kalman filter gain is given by

$$\mathbf{L} = \begin{bmatrix} -5.1e-1 & 2.6e+1 & 8.7e-1 & 1.6e+1 & 1.4e+1 & 1.9 & 2.6 & -8.3e-1 & 1.4e+1 & 5.1e-1 \\ 1.6e+1 & -1.6e+1 & -2.5e+1 & -5.1e-1 & 1.4e+3 & 6.9e+1 & -3.2e+2 & 1.8 & 5.1e-1 & -1.4e+1 \end{bmatrix}^T$$

and the corresponding Kalman filter poles are

$$s = -18.4 \pm 18.4, -18.4 \pm 18.5, -14.1 \pm 0.72, -0.86 \pm 0.50, -0.87 \pm 0.49.$$

4.8.7   Regulator Design

A regulator is designed next, assuming full-state feedback. The resulting regulator is combined with the Kalman filter from the previous step to form an LQG controller for the system. The regulator is based on LQR methodology, and the following weighting matrices are selected for the LQR gain calculations so that the corresponding LQG singular value plot (Figure 4.30) satisfies the singular-value frequency domain requirements:

$$\mathbf{Q} = diag\left(1,1,1,1,1,1,1,1,1\times10^8,1\times10^8\right) \text{ and } \mathbf{R} = \mathbf{I},$$
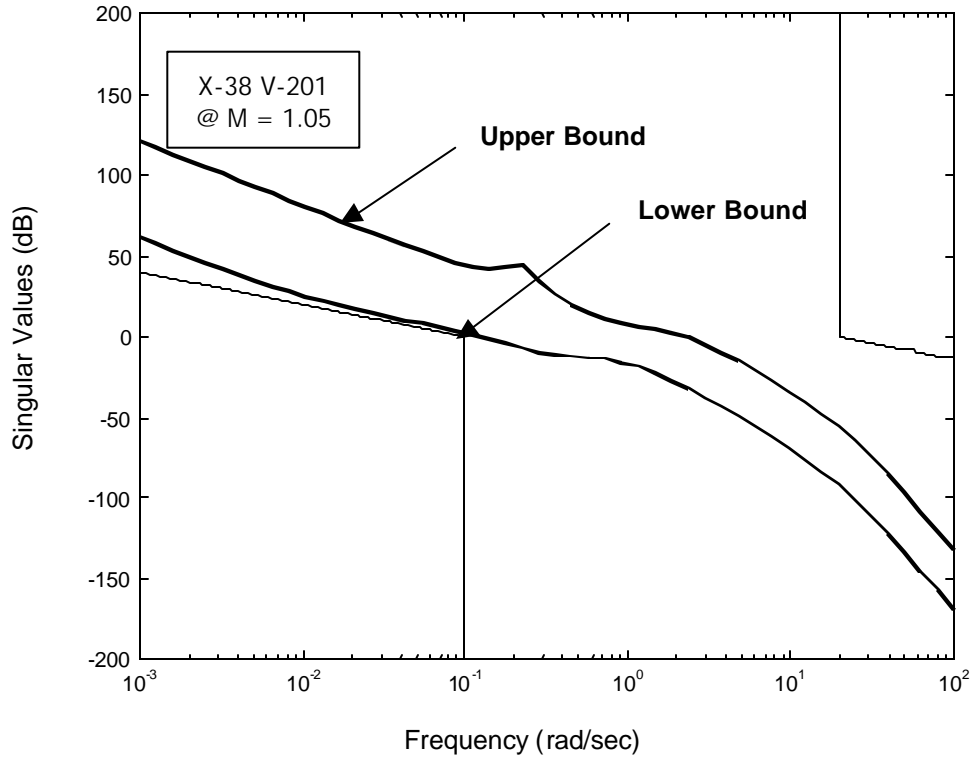

Figure 4.30 Singular values of the LQG regulator

The resulting regulator gain is as follows:

$$\mathbf{K} = \begin{bmatrix} -1.1e-1 & 8.2 & 1.2 & 3.0e-3 & 3.8e-2 & 1.4 & 3.9e-2 & 1.4 & 1.0e+4 & 1.7e-4 \\ -4.1e-2 & 1.2 & 3.5e+1 & -1.1e-1 & -4.3e-3 & -2.1e-1 & -4.2e-2 & -1.5 & 1.7e-4 & 1.0e+4 \end{bmatrix}^T \quad (4.35)$$

For the LQG singular value plot, the complete dynamics are given by the following augmented system:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\hat{\mathbf{x}}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{BK} \\ \mathbf{0} & \mathbf{A}-\mathbf{LC}-\mathbf{BK} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \hat{\mathbf{x}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ -\mathbf{L} \end{bmatrix} \mathbf{e}$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \hat{\mathbf{x}} \end{bmatrix}$$

(4.36)

with transfer function $\mathbf{CF}(\mathbf{s})\mathbf{BKF}_L(\mathbf{s})\mathbf{L}$. The singular values are plotted in Figure 4.30. The singular values of the resulting system clearly meet frequency domain specifications at all frequencies of interest.

4.8.8   Time Domain Analysis

Once the controller is designed to meet the frequency domain specification, its performance must be tested in the time domain against the time-domain specification (Figure 4.21). Figure 4.31 shows the response to a 10° bank angle step input to the system as defined in Equation (4.36). The bank angle response is within the design envelope while the sideslip angle is negligible. Also, both control surfaces are relatively inactive, and both displacement and rate are within the limits for each surface. The controller design, therefore, satisfies time-domain specifications at this stage.

4.8.9   Gain Scheduling Issues

As previously discussed, the main advantage over classical methods for the DI design methodology is the little need associated with the DI design methodology to schedule gains. In order to verify this claim, the designed controller, which is tuned at the transonic flight condition (Table 4.5, Case B), is applied to other flight conditions (Table 4.5, Cases A and C) without modification. The other two flight conditions, subsonic and supersonic, illustrate this analysis. The same 10° bank angle step inputs are applied to all three cases, and the resulting responses are presented in Figure 4.32 for the subsonic and transonic flight condition cases.
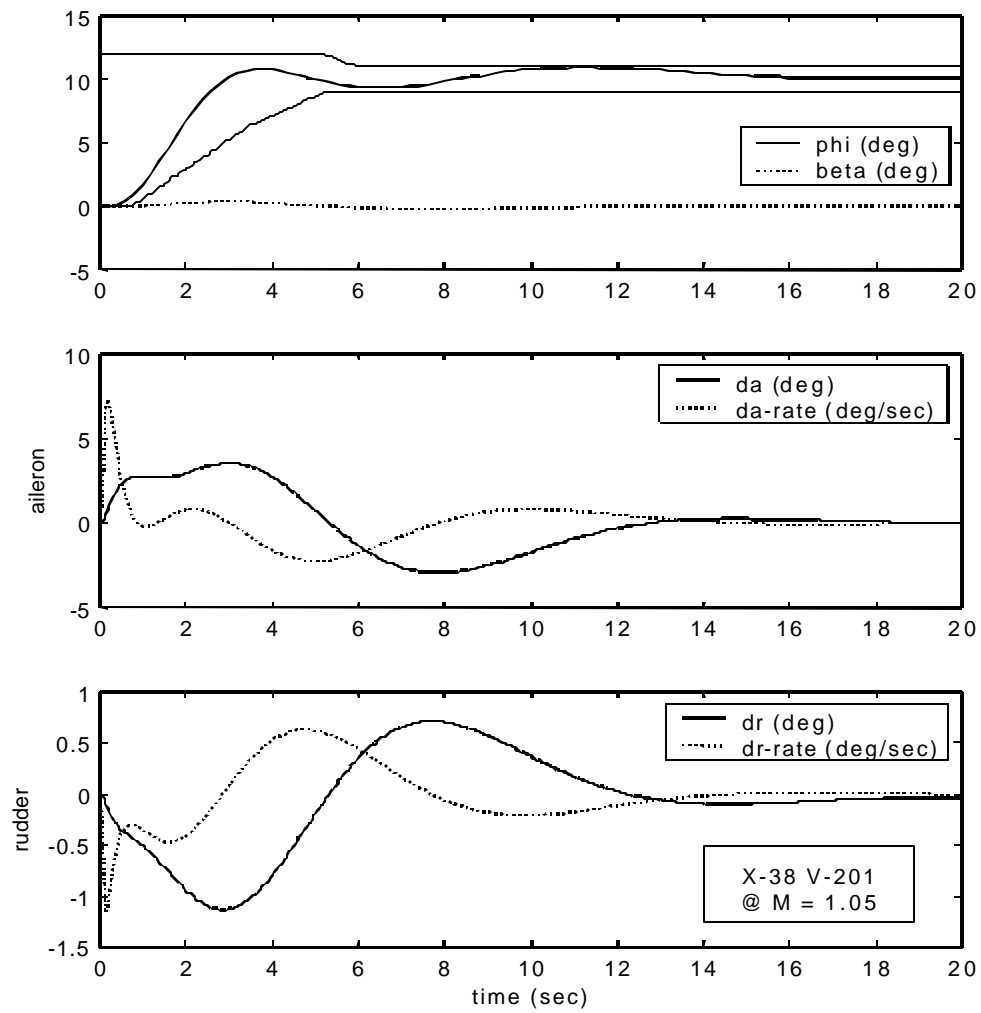
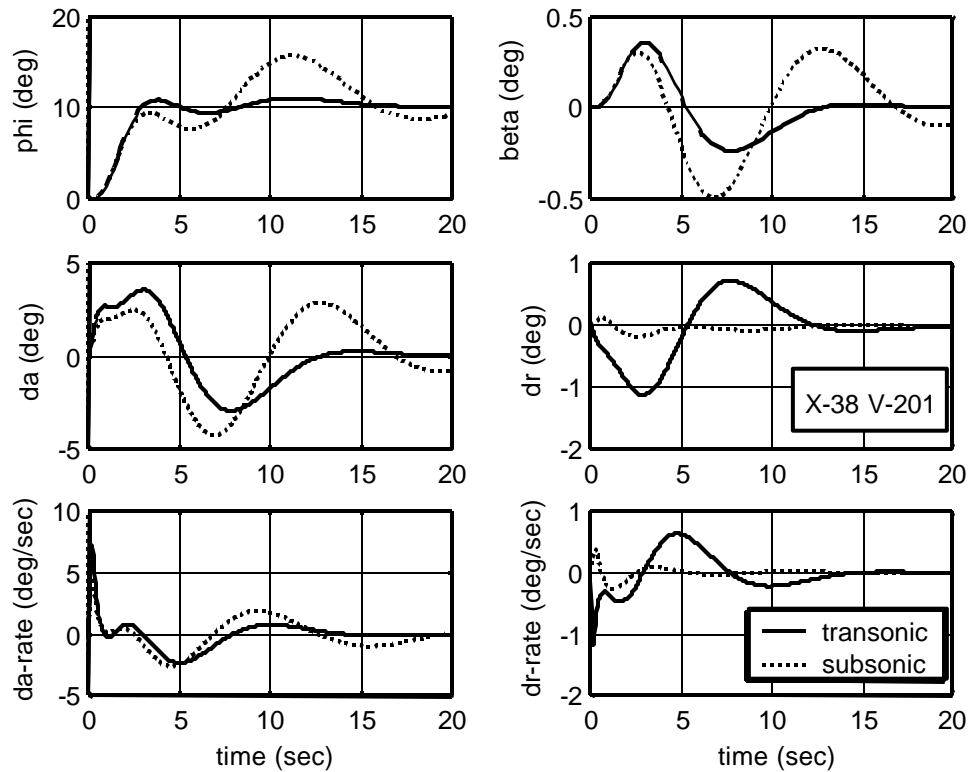*Figure 4.31 10° bank angle step response*

Figure 4.32 10° bank angle step response for different flight conditions

In the transonic and subsonic cases, the designed controller is able to stabilize the system. However, it fails to stabilize the system for the supersonic flight condition. In transonic and subsonic flight, control surface activities are well below the limits in rates and displacement. Though the objective of the DI controller is to produce a desired response at all flight conditions, no constraints are imposed on control surface activities in the DI control equation to achieve this. Control surface deflections and rates are based solely on the control distribution matrix. In order to avoid actuator saturation, the DI controller must command no more deflection or rate than the system hardware can provide or the system becomes nonlinear and the linear analysis may break down. But, these constraints overly restrict the available control power in some cases. Since a controller selected with this process may not produce the "best" performance at all flight conditions, the "nominal" (design) condition must be selected carefully to achieve good performance in a wide range of flight conditions. Moreover, the specification will change for different flight conditions. For example, the time-domain specification changes drastically near Mach = 1. Changing requirements for the entire flight envelope must be considered during the design process. The proposed approach allows an engineer to address this issue rather intuitively.

5 Robustness Analysis

5.1 μ-Analysis Applied to the X-38

5.1.1 Introduction

In the introductory sections of this document, singular-value decomposition is mentioned as the most common approach to adding robustness to a DI controller design. But it should be again emphasized that μ-analysis is by no means the only way to tackle robustness issues with this flight control design. Dang Vu[28] suggests that combined techniques using linear quadratic design, quantitative feedback theory, Lyapunov synthesis, adaptive control, and differential games have all been initiated together with μ-analysis.

5.1.2 Robustness Example: Application to the X-38 Lateral-Directional Aircraft Equations of Motion

The linear fractional transformation (LFT) structure – spelled out in detail in Section 6.2 of this document – is now applied to the lateral directional aircraft dynamics for the X-38. First, the lateral-directional aircraft equations of motion are represented as follows:

$$
\begin{bmatrix} \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \dfrac{Y_\beta}{V_T} & \dfrac{Y_p}{V_T} & \left(\dfrac{Y_p}{V_T}-1\right) & \dfrac{g}{V_T} \\ L_\beta & L_p & L_r & 0 \\ N_\beta & N_p & N_r & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \beta \\ p \\ r \\ \phi \end{bmatrix} + \begin{bmatrix} 0 & \dfrac{Y_{\delta_r}}{V_T} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix}
\tag{5.1}
$$

Next, we assume that – except for L_p, $L_{\delta a}$, N_r, and $N_{\delta r}$ – all parameters are certain values. We also know the variations (boundaries of uncertainty) for these parameters, too; and we will write them in the following form:

$$
\begin{aligned}
L_p^- &\leq L_p \leq L_p^+ \\
L_{\delta_a}^- &\leq L_{\delta_a} \leq L_{\delta_a}^+ \\
N_r^- &\leq N_r \leq N_r^+ \\
N_{\delta_r}^- &\leq N_{\delta_r} \leq N_{\delta_r}^+
\end{aligned}
\tag{5.2}
$$

These bounded uncertainties are now integrated into the aircraft equations of motion using the LFT form. Compare the decoupled roll axis equation

$$
\begin{aligned}
\dot{p} &= L_p p + L_{\delta_a} \delta_a \\
y &= p
\end{aligned}
\tag{5.3}
$$

with the general expression in Equation (6.8). Then, the following correspondences are clearly found:

$$x = p$$
$$u = \delta_a$$
$$a = L_p$$
$$b = L_{\delta_a}$$

(5.4)

Now, ensure the state space representation of the roll axis eqation with bounded uncertainties in L_p and L_{δ_a} is written by substituting the preceding relationship into Equation (6.13).

$$
\begin{bmatrix} \dot{p} \\ z_1 \\ z_2 \\ p \end{bmatrix} =
\begin{bmatrix}
L_p^{nom} & k_{p_1} & k'_{p_1} & L_{\delta_a}^{nom} \\
1 & k_{p_2} & 0 & 0 \\
0 & 0 & k'_{p_2} & 1 \\
1 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix} p \\ v_1 \\ v_2 \\ \delta_a \end{bmatrix}
$$

(5.5)

with

$$
k_{p_1} = \frac{2\left(L_p^+ - L_p^{nom}\right)\left(L_p^{nom} - L_p^-\right)}{\left(L_p^+ - L_p^-\right)}, \qquad
k_{p_2} = \frac{\left(L_p^+ - L_p^-\right) - 2L_p^{nom}}{\left(L_p^+ - L_p^-\right)}
$$

$$
k'_{p_1} = \frac{2\left(L_{\delta_a}^+ - L_{\delta_a}^{nom}\right)\left(L_{\delta_a}^{nom} - L_{\delta_a}^-\right)}{\left(L_{\delta_a}^+ - L_{\delta_a}^-\right)}, \qquad
k'_{p_2} = \frac{\left(L_{\delta_a}^+ - L_{\delta_a}^-\right) - 2L_{\delta_a}^{nom}}{\left(L_{\delta_a}^+ - L_{\delta_a}^-\right)}
$$

Similarly, yaw axis aircraft equation

$$\dot{r} = N_r r + N_{\delta_r} \delta_r$$
$$y = r$$

(5.6)

is put into state space form with uncertainties in N_r and N_{δ_r}.

$$
\begin{bmatrix} \dot{r} \\ z_3 \\ z_4 \\ r \end{bmatrix} =
\begin{bmatrix}
N_r^{nom} & k_{r_1} & k'_{r_1} & N_{\delta_r}^{nom} \\
1 & k_{r_2} & 0 & 0 \\
0 & 0 & k'_{r_2} & 1 \\
1 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix} r \\ v_3 \\ v_4 \\ \delta_r \end{bmatrix}
$$

(5.7)

with

$$k_{r_1} = \frac{2\left(N_r^+ - N_r^{nom}\right)\left(N_r^{nom} - N_r^-\right)}{\left(N_r^+ - N_r^-\right)}, \qquad k_{r_2} = \frac{\left(N_r^+ - N_r^-\right) - 2N_r^{nom}}{\left(N_p^+ - N_p^-\right)}$$

$$k_{r_1}' = \frac{2\left(N_{\delta_r}^+ - N_{\delta_r}^{nom}\right)\left(N_{\delta_r}^{nom} - N_{\delta_r}^-\right)}{\left(N_{\delta_r}^+ - N_{\delta_r}^-\right)}, \qquad k_{r_2}' = \frac{\left(N_{\delta_r}^+ - N_{\delta_r}^-\right) - 2N_{\delta_r}^{nom}}{\left(N_{\delta_r}^+ - N_{\delta_r}^-\right)}$$

Integrating both roll and yaw axis equations with Equation (5.1) yields the following LFT form for the lateral-directional equations:

$$\begin{bmatrix} \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \dfrac{Y_\beta}{V_T} & \dfrac{Y_p}{V_T} & \dfrac{Y_r}{V_T}-1 & \dfrac{g}{V_T} \\ L_\beta & L_p & L_r & 0 \\ N_\beta & N_p & N_r & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \beta \\ p \\ r \\ \phi \end{bmatrix} + \begin{bmatrix} 0 & \dfrac{Y_{\delta_r}}{V_T} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix}$$

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ \beta \\ p \\ r \\ \phi \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta \\ p \\ r \\ \phi \end{bmatrix} + \begin{bmatrix} k_{p_2} & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{p_2}' & 0 & 0 & 1 & 0 \\ 0 & 0 & k_{r_2} & 0 & 0 & 0 \\ 0 & 0 & 0 & k_{r_2}' & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ \delta_a \\ \delta_r \end{bmatrix}$$

(5.8)

     The input/output description of the plant and the uncertainty block are shown in Figures 5.1 and 5.2.
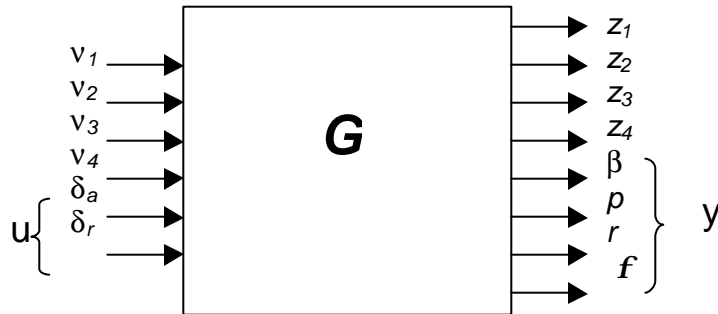

Figure 5.1 Plant input/output

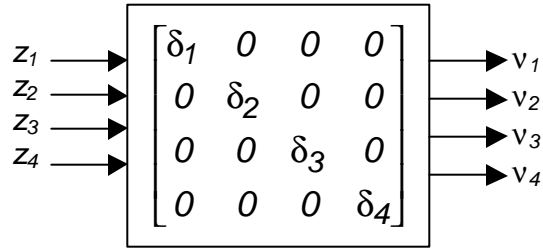*Figure 5.2 Uncertainty block*

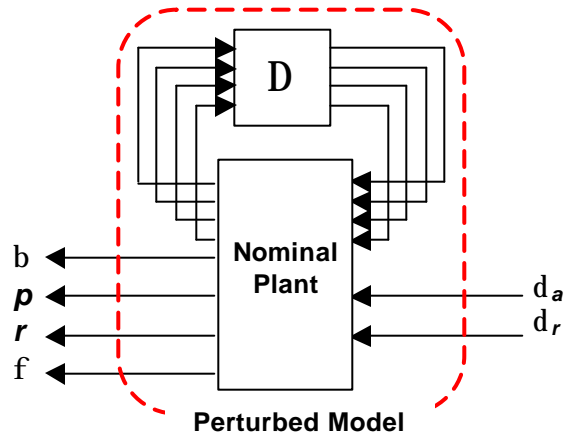These two blocks are interconnected to form a Parametric Uncertainty block, as shown in Figure 5.3.

Figure 5.3 Aircraft plant with parametric uncertainty

Parametric uncertainties are not the only type of uncertainties to be considered in this example. As discussed in Section 6.2.2.1, unmodeled dynamics or uncertainty at the input is another important type of uncertainty to be examined. When this is applied to our case study, the following weighting function, W_{in}, is used. Its place in the lateral-directional block diagram is shown in Figure 5.4.

$$W_{in} = \begin{bmatrix} \dfrac{s+1}{s+100} & 0 \\ 0 & \dfrac{s+1}{s+100} \end{bmatrix}$$
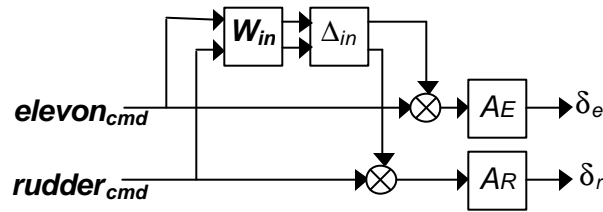(5.9)

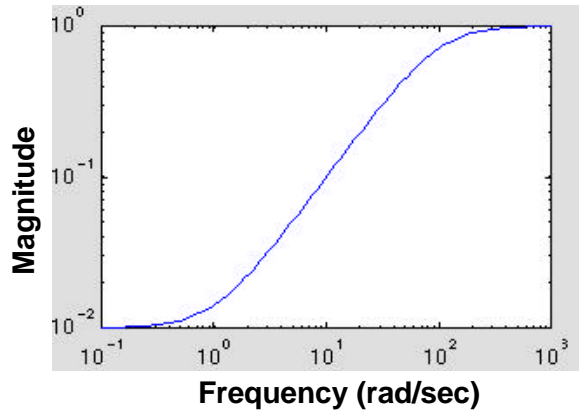*Figure 5.4 Unmodeled lateral-directional aircraft dynamics*

Figure 5.5 Uncertainty weighting function

This particular uncertainty weight, W_{in}, indicates that there is potentially 1% modeling error at low frequency and that the uncertainty in the model grows up to 100% at high frequency. The uncertainty weight is diagonal in form with equal diagonal elements. The perturbation mode is a circle, or a sphere, around the nominal plant.

A third type of uncertainty, discussed in Section 6.2.2.2, is uncertainty at the output (uncertainty in the measurements used) in the feedback loop. For our lateral-directional example, Figure 5.6 shows a block diagram representing the treatment of this type of uncertainty. Since it is unstructured, it is a function of input frequency – as was the uncertainty at the input.
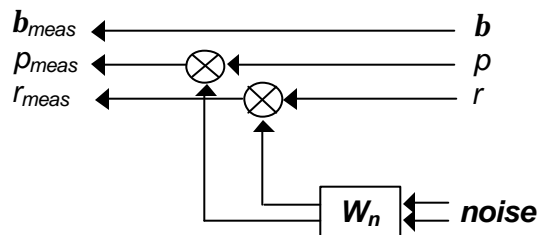

Figure 5.6 Unstructured uncertainty at the plant input due to output uncertainty
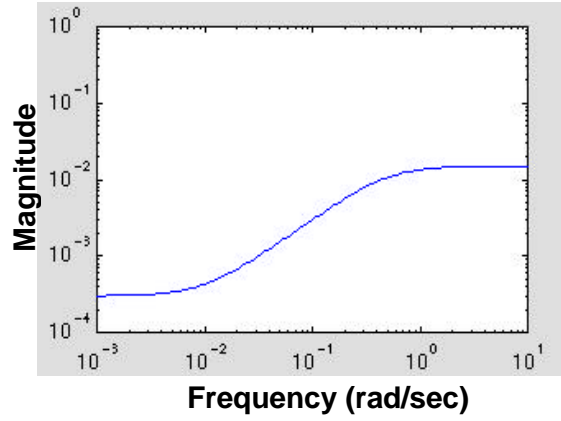
Figure 5.7 Unstructured output uncertainty weight

This output uncertainty weighing function implies in p and r a low-frequency measurement error of 0.003 rad/sec and a high-frequency measurement error of 0.015 rad/sec. The model of measured value of p, denoted p_{meas}, is given by

$$p_{meas} = p + W_{n_p}\eta_p, \tag{5.10}$$

where η_p is an arbitrary signal with

$$1 \geq \left\|\eta_p\right\|_2. \tag{5.11}$$

Any type of controller could be used in conjunction with our DI controller for μ-analysis. Much of the literature favors an H_∞ controller that can provide the robustness sought through μ-analysis – largely because μ-synthesis, which is an extended and more complex form of μ-analysis, requires an H_∞ controller combined with μ-analysis to synthesize an optimized controller to achieve stability performance. In our example, an H_∞ controller is designed using the MATLAB μ-Analysis and Synthesis toolbox[11].

The objective of the H_∞ controller is to make the pitch and yaw rate of a vehicle closely follow the commanded values of these two parameters (i.e., the goal is to minimize the errors e_p and e_p). Frequency-dependent weights are connected into the structure as shown in Figure 5.8.
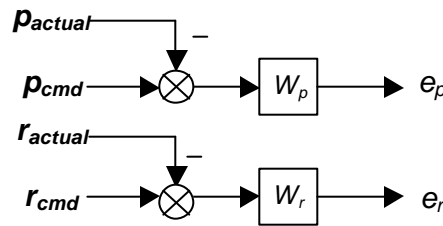

Figure 5.8 Performance Weighting block diagram

The overall performance error vector is given by

$$
e_{perf} = \begin{bmatrix} e_p \\ e_r \end{bmatrix} = \begin{bmatrix} \dfrac{s+10}{s+1} & 0 \\ 0 & \dfrac{s+10}{s+1} \end{bmatrix} \begin{bmatrix} p_{cmd} - p_{actual} \\ r_{cmd} - r_{actual} \end{bmatrix}. \tag{5.12}
$$

The shape of the weighting function (Figure 5.9) is chosen so that the controller provides performance in the low- to mid-frequency range. Error weights on the roll and yaw rates indicate a tolerance of 0.1 rad/sec at low frequency and 1 rad/sec at high frequency.
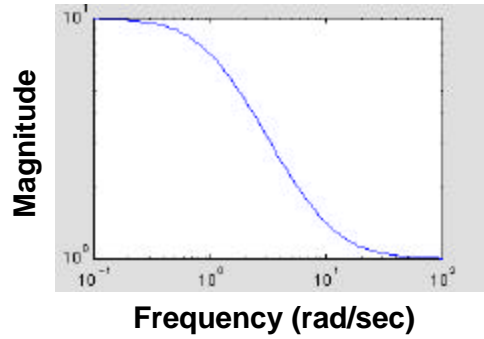
Figure 5.9 Performance weighting as a function of frequency

Limits on the actuator deflection magnitude and rates are also included in this example through W_{act} (actuator weight) shown in Figure 5.10. This mathematical constraint is not a physical "limit" but is treated as a constant weight matrix to produce the "error at the actuator" e_{act}.
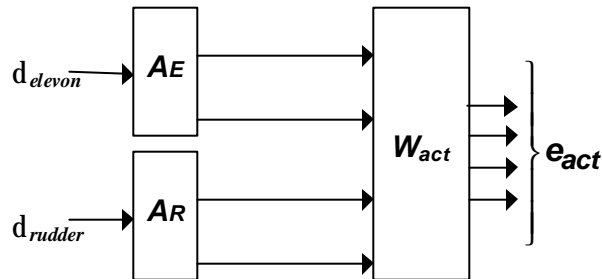

Figure 5.10 Control Surface Actuator Weights block diagram

This error vector is defined as

$$e_{act} = W_{act} \begin{bmatrix} \overset{\&}{\theta}_{elevon} \\ \theta_{elevon} \\ \overset{\&}{\theta}_{rudder} \\ \theta_{rudder} \end{bmatrix} \quad \text{with} \quad W_{act} = \begin{bmatrix} \dfrac{1}{50} & 0 & 0 & 0 \\ 0 & \dfrac{1}{22.5} & 0 & 0 \\ 0 & 0 & \dfrac{1}{50} & 0 \\ 0 & 0 & 0 & \dfrac{1}{25} \end{bmatrix}. \tag{5.13}$$

The two types of performance weights just described are combined to form a closed-loop weighted performance transfer matrix as shown in Figure 5.11.
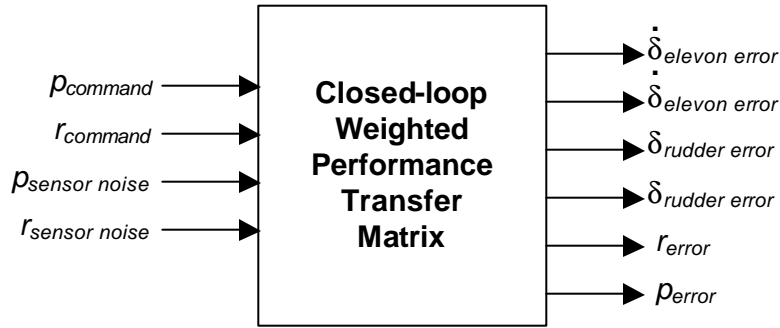
Figure 5.11 Weighted performance objective transfer matrix

Next, an H_∞ controller is designed to minimize the H_∞ norm of the nominal closed-loop transfer function from the disturbances to the errors. This design is actually a sub-optimal H_∞ controller because the controller is formulated after assuming that there is no model uncertainty. The *D-K* iteration process (part of the singular value decomposition process) must be done to approach a true optimal H_∞ controller. The input/output relation of the postulated sub-optimal H_∞ controller is shown in Figure 5.12.

Figure 5.12 H_∞ controller input/output

Although the designed system is a simple system, the order of the controller is 14. High order is a disadvantage of H_∞ controllers because system complexity increases as the order of the controller goes up.
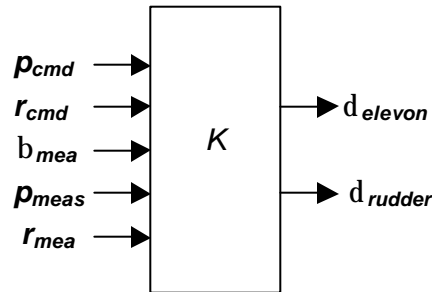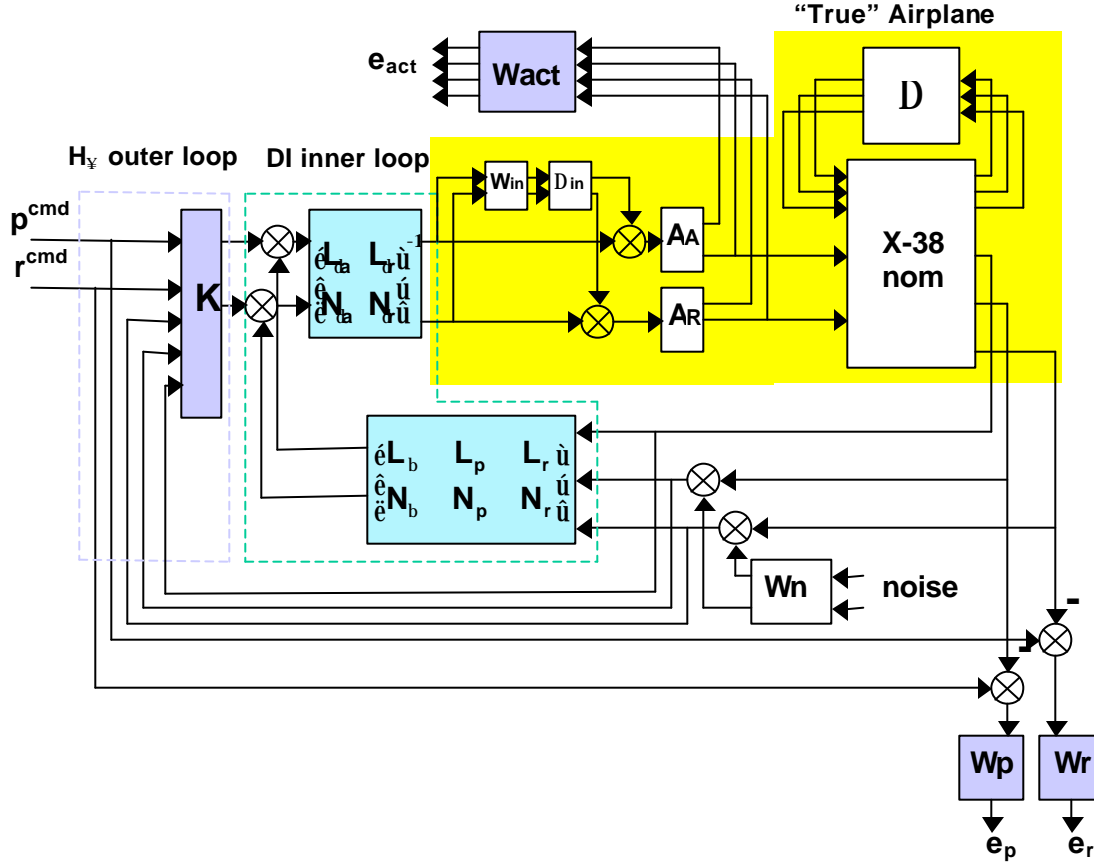

Figure 5.13 Interconnection structure

Subsystems developed in the previous sections are now integrated to form an interconnection structure for μ-analysis. A block diagram of the interconnection structure used for the analysis is given in Figure 5.13. The main objective in creating an interconnection structure is to transform all of the subsystem LFTs to a single (large) LFT that will separate the unknown parts from the known parts of the system. The interconnection structure can be formed relatively straightforwardly since standard linear operations – such as cascade connections, parallel connections, feedback connections, inversion, and frequency response – retain the LFT form. In other words, interconnections of LFTs are still LFTs. Since interconnection of the subsystem LFTs is a straightforward, albeit rather tedious, task, using commercially available software (e.g., the MATLAB μ-Synthesis Toolbox) is recommended.

Now, we are ready to apply the seven steps outlined in Section 6.2.3 to complete our μ-analysis and explore the stability robustness for our lateral-directional example. Applying the software tools mentioned above to the system shown in Figure 5.13, we complete the μ-analysis process. The flight condition simulated is the transonic portion of the flight envelope at $M_{\infty 0} = 1.05$, $\bar{q} = 222$ lb/ft^2, and $\alpha = 16.3°$. Uncertainty levels vary equally among four parameters, L_p, L_{δ_a}, N_r and N_{δ_r}, from 10% to 100%. Figure 5.14 summarizes the results obtained from this μ-analysis.
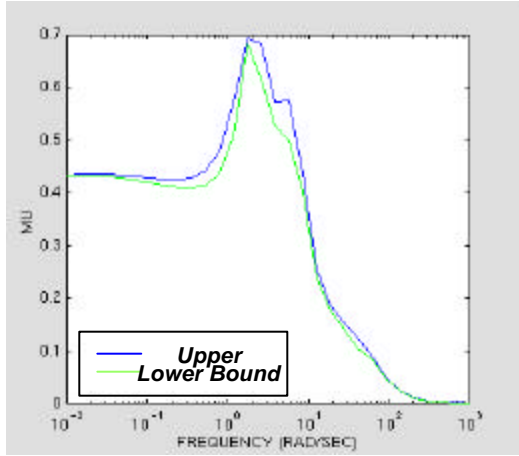
a) 10% Parametric Uncertainty
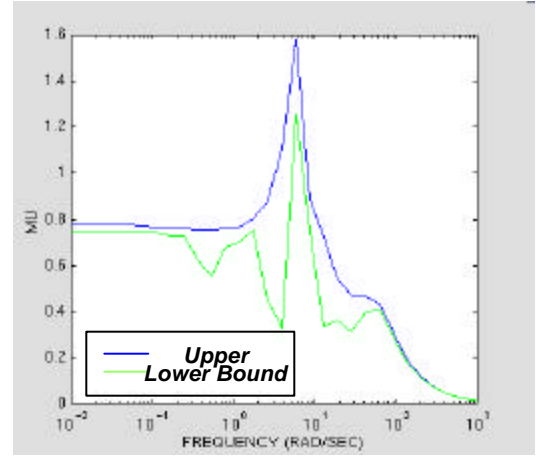
b) 25% Parametric Uncertainty
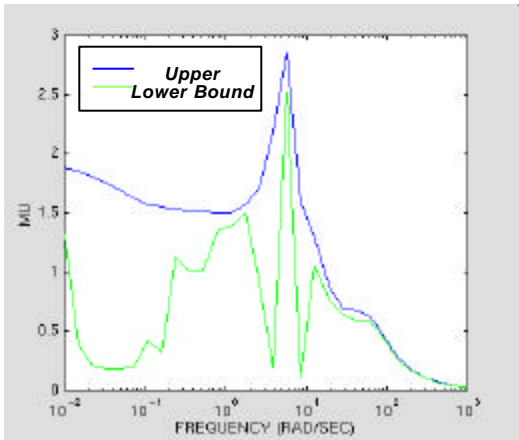
c) 50% Parametric Uncertainty
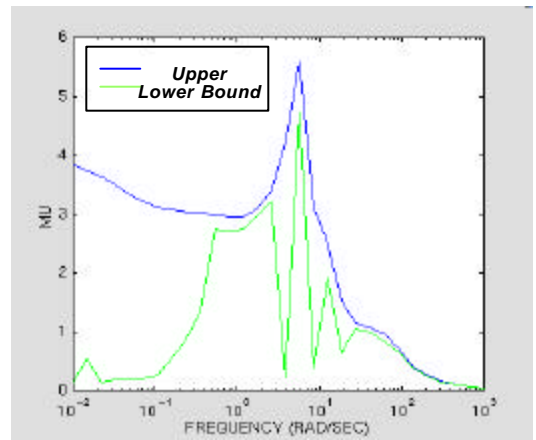
d) 100% Parametric Uncertainty

Figure 5.14 Parametric uncertainty results

The results show that as little as 15% parametric uncertainty causes a divergence from stability. The implication is that the DI controller is quite sensitive to parametric uncertainty.

Next, uncertainties are applied individually to the previously mentioned four coefficients. This iteration, on the μ-analysis procedure, shows that uncertainty in N_r produces the least system stability sensitivity; uncertainties in L_p lead to the most sensitivity. Figure 5.15 summarizes these results.
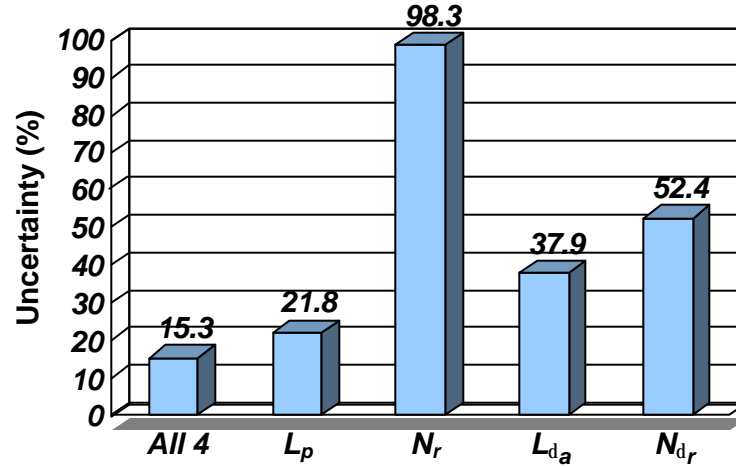
Figure 5.15 Maximum uncertainty tolerances for stability

5.2   Linear Quadratic Robustness Analysis Applied to the X-38

5.2.1   Introduction

A useful and easily understood performance index, J_w, is proposed by Ghaoui[29], et al. Their time domain performance index is simply the value of the usual linear quadratic performance index. However, Ghaoui, et al., have shown that when this performance index is used for worst-case analysis by taking the worst initial condition vector of unit magnitude, it is a simple, yet powerful metric for

1.   Comparing the performance of different controllers used with a given plant,
2.   Determining the worst disturbance histories for a given open- or closed-loop plant, and
3.   Determining the worst parameter changes for a given open- or closed-loop plant.

Using this performance index, the controller synthesized in Section 4.8 is analyzed for performance and robustness. First, the performance of the LQG controller is compared to that of full-state feedback; i.e., the LQR controller. Then, the same index is used to evaluate the robustness properties to parametric uncertainty as well as to sensor noise and external disturbances (in particular, side force gust). Analysis is extended to a nonlinear system with an LQR controller. Both the control surface positions and rates are allowed to saturate, and the guaranteed domain of stability is obtained. Finally, this nonlinear analysis is further extended to a simple control surface actuator failure analysis.

5.2.2   Performance Analysis

Given a stable and observable linear dynamic system,

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}, \quad \mathbf{x}(0) = \mathbf{x}_0,$$
$$\mathbf{y} = \mathbf{Cx}$$

(5.14)

J_w is defined as

$$J_w \equiv \max_{x_0} \frac{\int_0^\infty \mathbf{y}^T \mathbf{Q} \mathbf{y}\, dt}{\mathbf{x}_0^T \mathbf{x}_0}.\tag{5.15}$$

This is equivalent to

$$J_w = \max[l(\mathbf{S})],\tag{5.16}$$

where $\mathbf{S}$ is obtained by solving the steady-state Lyapunov equation

$$\mathbf{S}\mathbf{A} + \mathbf{A}^T \mathbf{S} + \mathbf{C}^T \mathbf{Q} \mathbf{C} = 0\tag{5.17}$$

and the state initial condition, $\mathbf{x}_0$, is the corresponding eigenvector

$$\mathbf{x}_0 = eigvec(\mathbf{S}).\tag{5.18}$$

The following index is used throughout the controller analysis:

$$J_w = \int_0^\infty \left(b^2 + f^2 + da^2 + d\dot{a}^2 + dr^2 + d\dot{r}^2 \right) dt\tag{5.19}$$

The value of this index is $J_{wLQG} = 1.25e+5$ for the designed LQG controller. If we assume full-state feedback (LQR controller), the index decreases to $J_{wLQR} = 2.93e+4$. Thus, the LQR controller is approximately four times better than the LQG controller using the performance index specified in Equation (5.19). The difference in performance comes from the fact that the LQG controller has to estimate unmeasured states using an observer, whereas the LQR controller uses all "perfectly measured" states for feedback.

The corresponding worst initial condition vector of unit length for the LQG controller is calculated as

$$\begin{aligned}\mathbf{X_0} &= \begin{bmatrix} b_0 & p_0 & r_0 & f_0 & d\dot{a}_0 & da_0 & d\dot{r}_0 & dr_0 & p_0^{des} & r_0^{des} \end{bmatrix}\\ &= [4.5e-3 \quad -3.3e-4 \quad -1.7e-4 \quad -3.2e-4 \quad -1.5e-6 \\ &\quad -5.3e-5 \quad -1.6e-6 \quad -5.4e-5 \quad 1.3e-1 \quad 9.9e-1]\end{aligned}\tag{5.20}$$

This worst initial condition vector indicates that performance is highly sensitive to the desired dynamics states, p_0^{des}, r_0^{des}. Among the vehicle states, sideslip angle is the most sensitive state to the overall performance.

5.2.3 Robustness Analysis – Parametric Uncertainties

One application of this time-domain quadratic performance index is to measure the performance of the system with parameter changes. Conceptually, this process first finds the worst direction in the parameter space and stretches parameter variations in this direction until the system becomes unstable

(i.e., $J_w \to \infty$). A brief summary of how this robustness criterion applies to a compensated system is paraphrased from the original work of Ghaoui[29], et al. Consider a linear system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \mathbf{x}(0) = \mathbf{x}_0,$$
$$\mathbf{y} = \mathbf{C}\mathbf{x} \tag{5.21}$$

with a dynamic compensator

$$\dot{\mathbf{x}}_c = \mathbf{A}_c\mathbf{x}_c + \mathbf{B}_c\mathbf{y}_s, \quad \mathbf{x}(0) = \mathbf{x}_0,$$
$$\mathbf{y}_s = \mathbf{C}_s\mathbf{x} \tag{5.22}$$

where the c subscript denotes controller values and the s subscript denotes measured properties. Then, the performance index is rewritten as

$$J_w = \max_{\mathbf{x}_0} \frac{\int_0^\infty \mathbf{y}^T\mathbf{Q}\mathbf{y}\,dt}{\mathbf{x}_0^T\mathbf{x}_0}, \tag{5.23}$$

which can be calculated easily from

$$J_w = \max(\mathbf{I}(\mathbf{S})) \tag{5.24}$$

where $\mathbf{S}$ is the submatrix in the solution of the steady-state Lyapunov equation

$$\mathbf{S}_a\mathbf{A}_a + \mathbf{A}_a^T\mathbf{S}_a + \mathbf{Q}_a = 0 \tag{5.25}$$

where

$$\mathbf{A}_a = \begin{bmatrix} \mathbf{A} & \mathbf{B}\mathbf{C}_c \\ \mathbf{B}_c\mathbf{C}_s & \mathbf{A}_c \end{bmatrix}, \quad \mathbf{Q}_a = \begin{bmatrix} \mathbf{C}^T\mathbf{Q}\mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_c^T\mathbf{R}\mathbf{C}_c \end{bmatrix} \quad \text{and} \quad \mathbf{S}_a = \begin{bmatrix} \mathbf{S} & (\cdot) \\ (\cdot) & \mathbf{S}_c \end{bmatrix}$$

with

$$\mathbf{x}_a = \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_c \end{bmatrix}$$

and $\mathbf{Q}$ and $\mathbf{R}$ are the performance weighting matrices that define the performance index in Equation (5.24). The a subscript denotes augmented properties. Then, the performance of the system with parametric variations is obtained as follows.

Let $\mathbf{p}$ be the vector of plant parameters of interest, where

$$\mathbf{A} = \mathbf{A}(\mathbf{p}), \quad \mathbf{B} = \mathbf{B}(\mathbf{p}), \quad \mathbf{C} = \mathbf{C}(\mathbf{p}). \tag{5.26}$$

The parameter vector, **p**, can be broken down into combinations of the nominal value of **p**, $\mathbf{p}_{nom}$, and the variation from the the nominal value, $\Delta \mathbf{p}$, as

$$\mathbf{p} = \mathbf{p}_{nom} + \Delta \mathbf{p} \,. \tag{5.27}$$

Then, a scalar measure of simultaneous changes in all parameters is defined as

$$s(\mathbf{p}) = \sqrt{(\Delta \mathbf{p})^T \mathbf{S}^{-2} \Delta \mathbf{p}} \tag{5.28}$$

where

$$\mathbf{S} = \text{diagonal matrix of standard deviation.}$$

Now, we can determine the $\Delta \mathbf{p}$ that maximizes J_w for a specified value of σ in Equation (5.28). Ghaoui[29], et al., have shown that a necessary condition for the maximum is

$$\Delta \mathbf{p} \equiv \mathbf{p} - \mathbf{p}_{nom} = s \cdot \mathbf{S} \cdot \mathbf{a} \tag{5.29}$$

where **a** is a unit vector in the direction of the gradient

$$J_p = \mathbf{S} \cdot \left(\frac{\partial J_w}{\partial \mathbf{p}} \right)^T, \tag{5.30}$$

which is evaluated at p and $\dfrac{\partial J_w}{\partial p_i}$ is calculated as

$$\frac{\partial J_w}{\partial p_i} = 2 \cdot tr \left[\mathbf{P} \left(\mathbf{C}^T \mathbf{Q} \frac{\partial \mathbf{C}}{\partial p_i} + \mathbf{S} \frac{\partial \mathbf{A}}{\partial p_i} \right) \right], \tag{5.31}$$

where **P** is determined by the Lyapunov equation

$$\mathbf{AP} + \mathbf{PA}^T + \mathbf{x_0}\mathbf{x_0^T} = \mathbf{0} \,. \tag{5.32}$$

Note that

$$\mathbf{P} = \int_0^\infty \mathbf{x}\mathbf{x}^T \, dt \tag{5.33}$$

This equation starts with the worst initial condition vector $\mathbf{x_0}$ (i.e., the initial condition that maximizes J_w).

Twelve parameters are selected for parametric uncertainty analysis. They form the vector of plant parameters, **p**, in Equation (5.27) as

$$\mathbf{p} = \begin{bmatrix} Y_b & Y_{dr} & L_b & L_p & L_r & L_{da} & L_{dr} & N_b & N_p & N_r & N_{da} & N_{dr} \end{bmatrix}^T \tag{5.34}$$

Then, the value of three standard deviations for each parametric uncertainty is assumed to be equal to the magnitude of the nominal value of each parameter.

For the designed LQG controller, the gradient of the performance index with respect to these 12 parameters was calculated as

$$
\frac{\partial J_w}{\partial \mathbf{p}} \cdot \mathbf{S} = \begin{bmatrix} -2.8e1 & 3.3 & -2.2e7 & 1.8e5 & 6.6e2 & -5.7e4 \\ 1.2e4 & -1.5e3 & 3.0e2 & 5.4e3 & -3.5e4 & 2.9e4 \end{bmatrix} . \tag{5.35}
$$

The magnitude of this gradient divided by the nominal J_w is 175, indicating that the performance index increases 175 times for a one-sigma change in the worst direction for the parameter space.

Figure 5.16 shows $1/J_w$ versus σ, where σ is defined in Equation (5.28) for the system with the LQG controller.

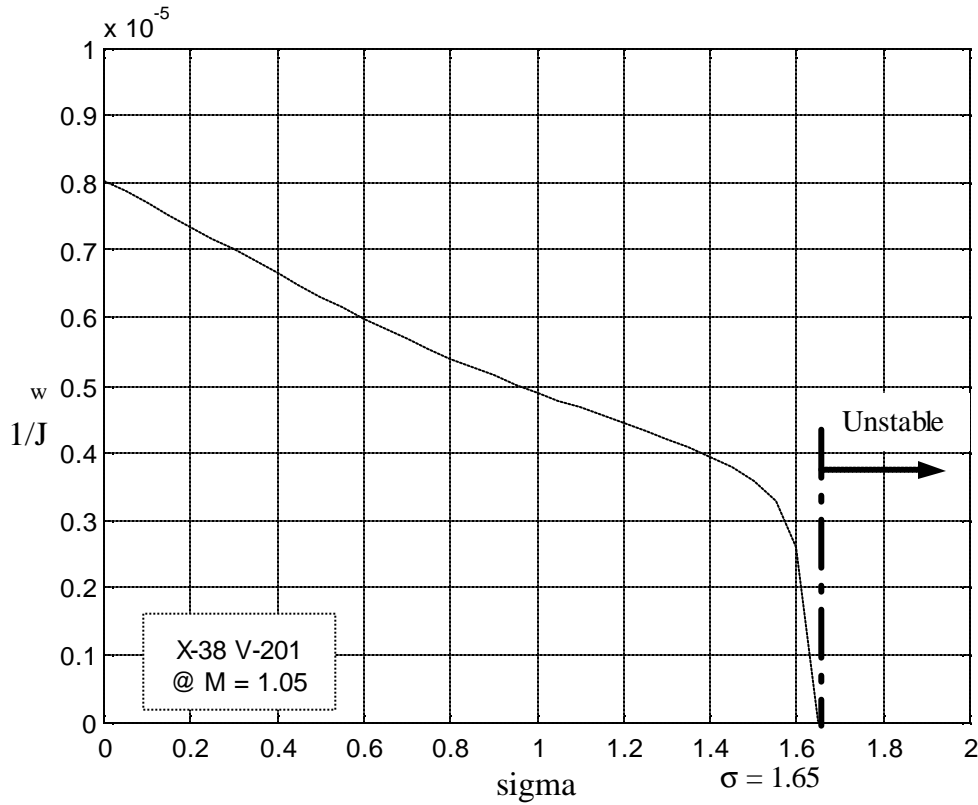***Figure 5.16 $1/J_w$ versus σ for worst parameter change***

System instability occurs at $\sigma = 1.65$ in the worst direction in the parameter space. At the stability parameter margin, i.e., $\sigma = 1.65$,

$$
\Delta p^T \cdot \mathbf{S}^{-1} = \begin{bmatrix} -2.1e-6 & 2.5e-7 & -1.7 & 1.3e-2 & 5.0e-5 & -4.3e-3 \\ 9.0e-4 & -1.1e-4 & 2.3e-5 & 4.1e-4 & -2.7e-3 & 2.2e-3 \end{bmatrix} . \tag{5.36}
$$

This indicates that the most important major contribution is from a decrease in L_β and, to a lesser degree, a decrease in L_p. The sensitivities of performance due to all other parameters are negligible as compared to these two parameters.

5.2.4   Robustness Analysis – Disturbance

So far, all performance analyses assume no disturbances are present. However, the presence of noise is inevitable in physical systems. In this section, the original performance criterion is modified so that both process noise, w, and measurement noise, v, are addressed in the performance index. To achieve this, the worst disturbances are assumed to be feedbacks of the augmented state, $\mathbf{x}_a$, where the gain matrices are determined by the solution of the Riccati equation

$$\mathbf{S}_a\mathbf{A}_a + \mathbf{A}_a^T\mathbf{S}_a + \mathbf{Q}_a + \frac{1}{h}\mathbf{S}_a\mathbf{G}_a\mathbf{R}_a^{-1}\mathbf{G}_a^T\mathbf{S}_a = 0 , \tag{5.37}$$

where $\mathbf{S}_a$, $\mathbf{A}_a$, and $\mathbf{Q}_a$ are defined in Equation (5.25) and

$$\mathbf{G}_a = \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_c \end{bmatrix}, \ \mathbf{R}_a = \begin{bmatrix} \mathbf{R}_w & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_v \end{bmatrix}, \ w = \mathbf{K}_w \cdot \mathbf{x}_a, \ v = \mathbf{K}_v \cdot \mathbf{x}_a \tag{5.38}$$

with

$$\begin{bmatrix} \mathbf{K}_w \\ \mathbf{K}_v \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{wx} & \mathbf{K}_{wc} \\ \mathbf{K}_{vx} & \mathbf{K}_{vc} \end{bmatrix} = \frac{1}{m}\begin{bmatrix} \mathbf{R}_w^{-1}\mathbf{G}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_v^{-1}\mathbf{B}_c^T \end{bmatrix}\mathbf{S}_a . \tag{5.39}$$

Now, J_w is given by

$$J_w = \max[\mathbf{l}(\mathbf{S})] + hW , \tag{5.40}$$

where h is a Lagrange multiplier and W is determined from

$$W = tr\left[\mathbf{P}_a\,\mathbf{K}_a^T\mathbf{R}_a\mathbf{K}_a\right] \tag{5.41}$$

and $\mathbf{P}_a$ is the solution to the Lyapunov equation

$$(\mathbf{A}_a + \mathbf{G}_a\mathbf{K}_a)\mathbf{P}_a + \mathbf{P}_a(\mathbf{A}_a + \mathbf{G}_a\mathbf{K}_a)^T + \mathbf{x}_{a0}\mathbf{x}_{a0}^T = \mathbf{0} \tag{5.42}$$

and

$$\mathbf{K}_a = \begin{bmatrix} \mathbf{K}_w \\ \mathbf{K}_v \end{bmatrix} . \tag{5.43}$$

First, the original system defined in Equation (4.31) is modified to accommodate a disturbance due to side force gust.

$$\dot{\mathbf{x}}_{dist} = \mathbf{A}_{dist}\mathbf{x}_{dist} + \mathbf{B}_{dist}\mathbf{u}_{dist} + \mathbf{G}_{dist}\mathbf{w}_{dist} \tag{5.44}$$

where

$$\mathbf{A}_{dist} = \mathbf{A}_{aug}, \quad \mathbf{B}_{dist} = \mathbf{B}_{aug}, \quad \mathbf{x}_{dist} = \mathbf{x}_{aug}, \quad \mathbf{u}_{dist} = \mathbf{u}_{aug}$$
$$\mathbf{G}_{dist} = \begin{bmatrix} -Y_b/V_T & -L_b & -N_b & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{5.45}$$
$$w_{dist} = b$$

The difference from the original system is the addition of the $\mathbf{G}_{dist}w_{dis}$ term. This formulation is interpreted as side force translated into the sideslip angle, and this disturbed sideslip angle acts as a control although it is trying to destabilize the system. In our example, we select max $|\beta| = 1$ degree for transonic flight.

Next, another modification to the original system is made to accomodate the measurement noise. Adding sensor errors to continuously changing scale factors and biases to the measurements vector is the technique used.

$$\mathbf{y}_s = \begin{bmatrix} f_s \\ b_s \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\mathbf{x}_{dist} + \mathbf{v} \tag{5.46}$$

Also, we assume the worst disturbances are feedbacks of the augmented state $\mathbf{x}_a$, where gain matrices are determined by the solution of the Riccati equation specified in Equation (5.37). In our example, $\mathbf{R}_w = 1$ and $\mathbf{R}_v = \text{diag}(3,1)$ are selected and the corresponding value of η is 6.84×10^4 by interpolation. Disturbances are given by the positive feedbacks defined in Equation (5.38); and the feedback sideslip disturbance vector, $\mathbf{K}_w$, and sensor noise matrix, $\mathbf{K}_v$, are calculated as follows:

$$\begin{aligned}
\mathbf{K}_w = [&-5.9e-3 \quad 2.9e-3 \quad 2.1e-2 \quad -1.0e-4 \quad 6.8e-6 \quad 2.5e-4 \quad -1.4e-5 \quad -5.1e-4 \quad -1.8e-4 \quad -1.8e-2 \\
& -2.3e-3 \quad -1.2e-3 \quad -1.2e-2 \quad 1.3e-3 \quad -1.3e-6 \quad -4.2e-5 \quad 9.9e-6 \quad 3.6e-4 \quad 7.8e-4 \quad 2.0e-2]
\end{aligned}$$
$$\tag{5.47}$$

$$\mathbf{K}_v = \begin{bmatrix} 2.0e-5 & -6.7e-6 & -1.8e-5 & -1.9e-6 & -2.5e-8 & -9.1e-7 & -9.1e-9 & -3.2e-7 & -6.3e-6 & -1.8e-5 \\ 1.4e-5 & 4.9e-5 & 1.5e-3 & -8.1e-5 & -2.7e-7 & -1.0e-5 & -1.8e-6 & -6.8e-5 & 8.8e-5 & -8.6e-4 \\ -5.0e-6 & -1.1e-7 & -5.4e-6 & -2.5e-6 & -1.3e-9 & -6.8e-8 & 8.6e-9 & 3.0e-7 & 3.3e-6 & 3.2e-6 \\ -3.8e-4 & -8.6e-6 & -8.8e-4 & 6.5e-5 & 2.8e-7 & 1.0e-5 & 1.1e-6 & 4.1e-5 & -5.5e-5 & 1.2e-3 \end{bmatrix}$$

The value of J_w increases to 1.69e+5, a 35% increase from the nominal case value of 1.25e+5. The worst unit initial condition vector is

$$\begin{aligned}
\mathbf{x}_0 = [&4.8e-3 \quad -2.9e-4 \quad -2.6e-4 \quad -3.4e-4 \quad -1.3e-6 \\
& -4.7e-5 \quad -1.3e-6 \quad -4.4e-5 \quad 1.3e-1 \quad 9.9e-1]^T.
\end{aligned} \tag{5.48}$$

The magnitude of the initial measurement noise is obtained by substituting Equations (5.47) and (5.48) into Equation (5.38). The ratio between the actual values to the noise for sideslip angle and bank angle is calculated as 5.4%, and 17.4%, respectively, at $t = 0$.

5.2.5 Domain of Stability for the System with Actuator Saturation

Although the time domain criterion we have chosen is a useful tool for controller robustness analysis, its major drawback is that the entire structure is defined using a linear system assumption. However, most aerospace systems are not linear; so this point is especially significant since the performance of a DI controller is sensitive to the available control power. This concern essentially arises because the control inputs, $\mathbf{u(x)}$, is proportional to the inverse of the magnitude of the control distribution function.

Therefore, the control surface position and rate should be included when the boundary of the stability region is considered. Although this is a significant weakness of DI methodology, relatively little research has come to our attention[30-32]. Recently, Tarbouriech[33], et al., published a technique to compute a guaranteed domain of stability for a system subject to position- and rate-limited system inputs. This study shows two different approaches: the algebraic Ricatti equation (ARE) approach and the linear matrix inequality (LMI) approach. Here, the simpler and more widely used approach of the two, the ARE approach, is used to find the domain of the system stability.

The objective is to find the largest possible Lyapunov stability parameter, $\mathbf{r}$, such that the closed-loop system is locally stable in the largest Lyapunov level set. In other words, we seek the largest domain of initial state vectors that produces a stable solution to the given Ricatti equation defined by Tarbouriech[33], et al. Though this methodology produces an optimal solution by solving a given Ricatti equation, this optimal solution depends on the choice of state and control weight matrices, which is always the case for Ricatti solutions. Therefore, the largest domain of initial condition predicted by this approach does not grantee the largest stable initial condition domain globally, and the computed stability domain may still be a conservative prediction.

Actuator position and rate limits were previously listed in Section 4.1.2. Using the DI formulation, the actuator dynamics are not accessible directly. Instead, the desired yaw and roll acceleration and their time derivatives are used to limit the control inputs. For the yaw axis, the desired yaw acceleration is bounded by

$$\left|\dot{r}^{des}\right|_{max} = N_{da}\left|da\right|_{max} + N_{dr}\left|dr\right|_{max} = 0.485\left(rad / \sec^2\right) \tag{5.49}$$

and the time derivative of the yaw acceleration is limited to

$$\left|\ddot{r}^{des}\right|_{max} = N_{da}\left|\dot{da}\right|_{max} + N_{dr}\left|\dot{dr}\right|_{max} = 0.990\left(rad / \sec^3\right). \tag{5.50}$$

Similarly, for the roll axis,

$$\left|\dot{p}^{des}\right|_{max} = L_{da}\left|da\right|_{max} + L_{dr}\left|dr\right|_{max} = 2.69\left(rad / \sec^2\right) \tag{5.51}$$

and

$$\left|\ddot{p}^{des}\right|_{max} = L_{da}\left|\dot{da}\right|_{max} + L_{dr}\left|\dot{dr}\right|_{max} = 5.66\left(rad / \sec^3\right). \tag{5.52}$$

Assuming full-state feedback with the regulator gain in Equation (4.39) and no parametric uncertainties and disturbances, the maximum value of the stability parameter ($\rho_{max} =$) is 14.5. When we consider the domain of stability in the two different states – bank and sideslip angles – simultaneously, a 3-D plot is obtained (Figure 5.17). The system is stable up to a 48.1° bank angle assuming no sideslip angle.

Similarly, the system is stable up to 1.59° sideslip assuming no bank angle. The system is not guaranteed to be stable outside of this domain.
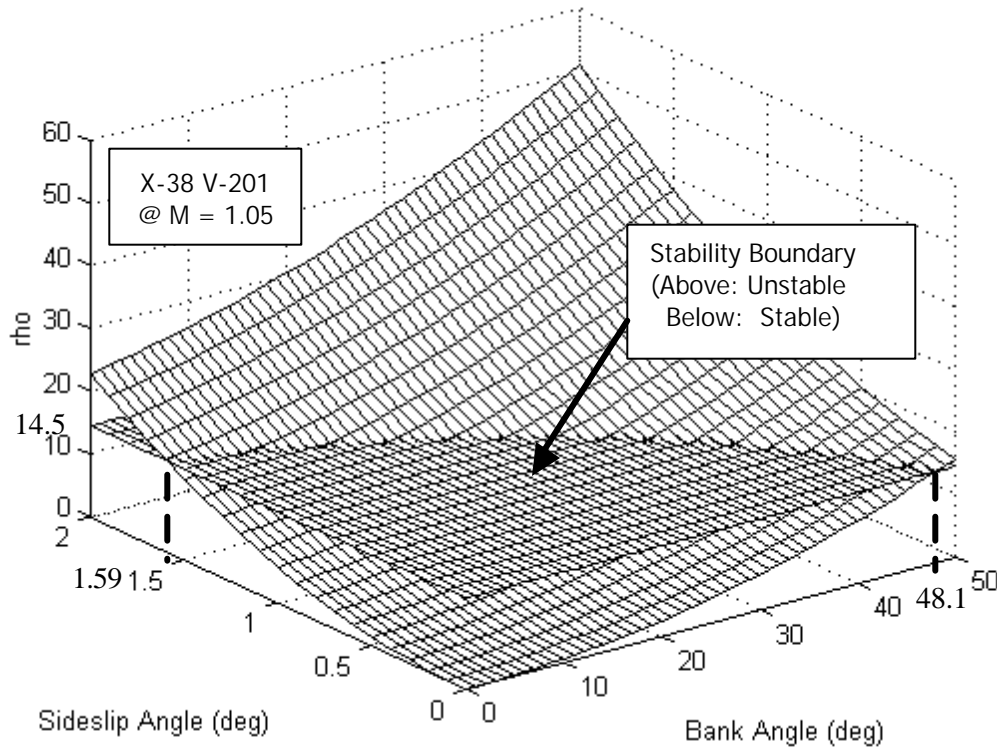

Figure 5.17 Stability boundary

5.2.6   Change in Domain of Stability due to Control Surface Actuator Failure

This nonlinear analysis next examines the stability domain in the event of a control surface actuator failure. This process changes the limits on the position and rates of the input vectors. Six failure modes are considerd, and results are compared to the nominal case where no control surface actuator failure has taken place. The nominal and failure mode cases that are considered are the: (1) nominal case, (2) one aileron failure case, (3) one rudder failure case, (4) one aileron and one rudder failure case, (5) two aileron actuators failure case, and (6) two rudder actuators failure case. Also, as in the previous section, perturbations on the inital conditions are limited to bank and sideslip angles. Figure 5.18 summarizes the results and their associated cross-sectional top view. This figure clearly illustrates the difference in stability Lyapunov levels due to different constraints on control inputs. As predicted, the nominal case, where none of the control surface actuators are failed (and the largest amount of control power is available), shows the highest level of stability. The ratio, compared to the nominal stability domain, for all six cases is summarized in Figure 5.19.
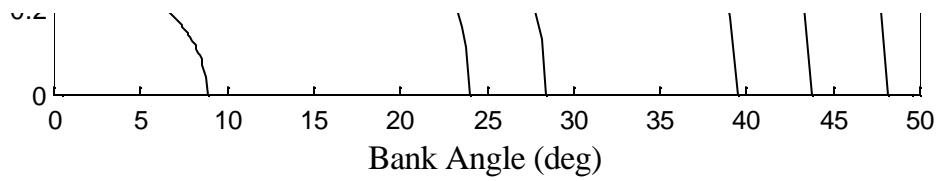
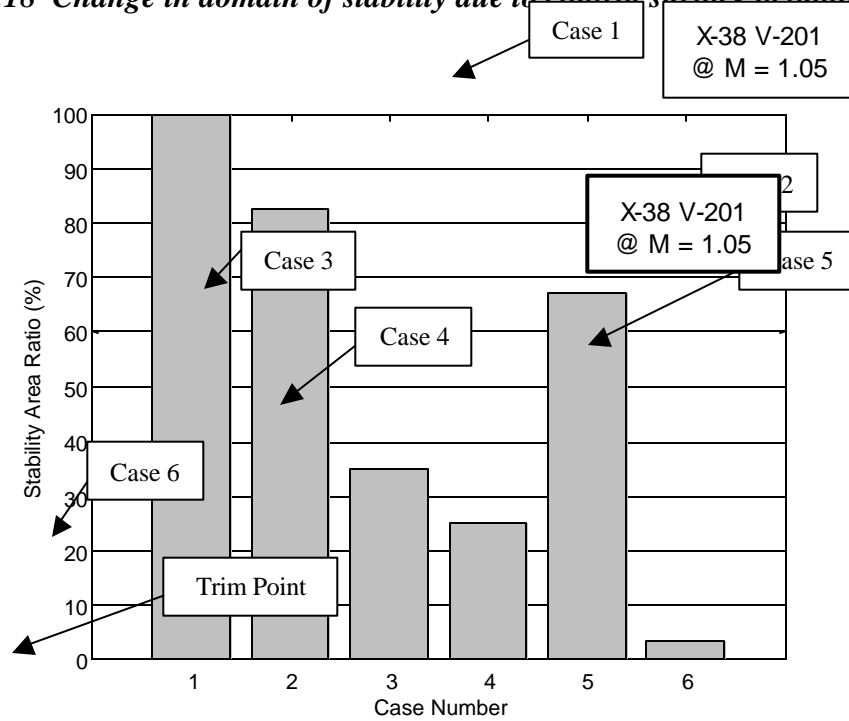**Figure 5.18 Change in domain of stability due to control surface actuator failure**

Figure 5.19 Area of stability comparison due to actuator failure

6   Theoretical Foundations

6.1   Basic Forms of Dynamic Inversion

Dang Vu asserts that the essentials of the DI approach are most easily understood in terms of an SISO system. Since we followed that approach in our learning process, as described in Section 3, we have postulated a class of linear systems affine[i] in control that is represented by the following mathematical form:

$$\dot{x} = f(x) + g(x)u \qquad (6.1)$$

$$y = h(x) \qquad (6.2)$$

where $f(x)$ and $g(x)$ are smooth vector fields on R^n and $h(x)$ is a smooth mapping function that maps R^n into R. This system is feedback linearizable of relative degree r if state and input transformations exist.

$$z = \Phi(x) \qquad\qquad z \in R^r$$

$$y = \alpha(x) + \beta(x)v \qquad v \in R$$

where $\beta(x) \neq 0$ and $\Phi(x)$ is a diffeomorphism[ii] that transforms Equation (6.1) into a controllable linear system.

$$\dot{z} = Az + Bv$$

Following Dang Vu's rather succinct development (with some clarifications), we differentiate the nonlinear output equation (Equation (6.2)) with respect to time and obtain

$$\dot{y} = \frac{\partial h}{\partial x}\dot{x} = \frac{\partial h}{\partial x}\left[f(x) + g(x)u\right] .$$

If the coefficient of u is zero, we continue with successive differentiations in the same fashion until a nonzero process coefficient surfaces. Dang Vu's succinct notation, which uses the Lie derivative from differential geometry for these repeated differentiations, is useful.

---

[i]"Affine in control" means that all transformations of finite system parameters remain finite under the controller's action.

[ii]A diffeomorphism is said to occur when the scalar components of a mapping (or transformation) F are r times differentiable with $r \geq 1$ with respect to the scalar components of x (with a discrete time mapping $x_k = F x_{k+1}$). The mapping must also be invertible; that is, $x_{k+1} = F^{-1} x_{k+1}$ must hold. Invertibility implies that F^{-1} exists. The scalar components of the inverse must likewise be r times differentiable.[34]

$$L_f(x) = \frac{\partial h}{\partial x} f(x) \tag{6.3}$$

Equation (6.3) is the Lie derivative of the scalar function h with respect to the vector field f. Higher-order derivatives have a similar form.

$$L_f^k h(x) = L_f\left(L_f^{k-1} h(x)\right) \tag{6.4}$$

Using this Lie derivative notation, the output equation can be rewritten as

$$\dot{y} = \frac{\partial h}{\partial x} \dot{x} = \frac{\partial h}{\partial x}\left[f(x) + g(x)u\right] = L_f h(x) + L_g h(x) u. \tag{6.5}$$

If the second derivative in Equation (6.5) is 0 – that is, $L_g h(x) = 0$ – a second differentiation yields

$$\ddot{y} = L_f^2 h(x) + L_g L_f h(x) u. \tag{6.6}$$

The differentiations end when $L_g L_f^{k-1} h(x) = 0$ for k =1, … , r – 1, but $L_g L_f^{r-1} h(x) \neq 0$. The last derivative of the sequence for the output response is then

$$y^{(r)} = L_f^r h(x) + L_g L_f^{r-1} h(x) u. \tag{6.7}$$

The number r is called the relative degree of the original control equation (Equation (6.1)).

If we define our coordinate transformation in terms of the Lie derivative,

$$z_k = \Phi_k(x) = L_f^{k-1} h(x) \quad \text{for } k = 1, 2, \dots, r,$$

the resulting transformed set of equations is linear, of dimension r, and in a companion form called the Brunovsky canonical form.

$$\dot{z} = \begin{bmatrix} 0 & 1 & 0 & \mathrm{L} & 0 \\ 0 & 0 & 1 & \mathrm{L} & 0 \\ 0 & 0 & \mathrm{L} & 1 & 0 \\ 0 & 0 & \mathrm{L} & 0 & 1 \\ 0 & 0 & \mathrm{L} & 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mathrm{M} \\ 1 \end{bmatrix} v = Az + Bv \tag{6.8}$$

where $v = L_f^r h(x) + L_g L_f^{r-1} h(x) u$. Vu points out that exact linearization is possible when the relative degree r is equal to the order of the system n and the linearized system (Equation (6.8)) is both controllable and observable.

Since we are interested in obtaining for the linearized system a control law that will impose desired behavior on the original nonlinear system, it is also necessary that we carefully examine any control law in terms of stability of the nonlinear system. Obviously, that requirement suggests that we

transform any postulated control law back into the original coordinates. Symbolically, we can write this transformation as

$$u = \alpha(x) + \beta(x)v$$

with
$$\alpha(x) = -\frac{L_f^r h(x)}{L_g L_f^{r-1} h(x)} \text{ and } \beta(x) = \frac{1}{L_g L_f^{r-1} h(x)}.$$

6.2   Stability and Robustness Analyses

In this section, stability and robustness analyses are described for the DI controller. The most commonly used methodology used to analyze robustness of linear systems using DI controllers is based on the structured singular value (μ) and a technique now widely described in the literature as μ-analysis. However, μ-analysis is not the only method used to examine the stability and robustness of DI-based controllers.

6.2.1   Linear Fractional Transformations

LFTs are used to integrate parameter variations (uncertainty) into the system under μ-analysis. As the first step in the μ-analysis procedure, all parameter variations are collected into an uncertainty matrix in LFT form. Then, μ-analysis looks for the smallest variation in these parameters that drive the system to instability. In this section, the methodology to create an LFT form is explained.

Suppose we have a linear system that is described by the following:

$$\dot{x} = ax + bu$$
$$y = x$$
(6.9)

Now, we assume that the value of a varies between a^- and a^+.

$$a^- \leq a \leq a^+$$
(6.10)

where

$$a^- = lowerlimit \ of \ variation \ in \ a$$
$$a^+ = upper \ limit \ of \ variation \ in \ a$$

This relation can be rewritten in terms of nominal value of a, a^{nom}, as

$$a = a^{nom} + \frac{k_1 \delta_a}{1 - k_2 \delta_a},$$
(6.11)

where

$$a^{nom} = \textit{nominal value of } a$$

$$k_1 = \frac{2\left(a^+ - \bar{a}\right)\left(\bar{a} - a^-\right)}{a^+ - a^-}$$

$$k_2 = \frac{\left(a^+ + a^-\right) - 2\bar{a}}{a^+ - a^-}$$

$$-1 \leq \delta_a \leq 1$$

When the upper and lower variations from the nominal values are equal – that is, when $\left| a^+ - a^{nom} \right| = \left| a^- - a^{nom} \right|$ – then the previous equation simplifies to

$$a = a^{nom} + k\delta_a, \tag{6.12}$$

where

$$a^{nom} = \textit{nominal value of } a$$

$$k = a^+ - a^{nom} = a^{nom} - a^-$$

$$-1 \leq \delta_a \leq 1$$

The perturbation in a described in Equation (6.11) is integrated into Equation (6.9) and is now expressed in state-space form by introducing the fictitious terms z_a and w_a as

$$\begin{bmatrix} \dot{x} \\ z_a \\ y \end{bmatrix} = \begin{bmatrix} \bar{a} & k_1 & b \\ 1 & k_2 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ w_a \\ u \end{bmatrix}. \tag{6.13}$$

The block diagram in Figure 6.1 (above) corresponds to the state-space expression (Equation (6.13)). Here, we have completely separated what is known, the G matrix, from what is uncertain, Δ. We treat Δ as uncertain; but, we do know its range: $-1 \leq \Delta \leq 1$. For now, uncertainty in a alone is considered. The next case to be considered is the case when there is an uncertainty in b. This case, variation in b, is expressed similar to the previous case as

$$\begin{bmatrix} \dot{x} \\ z_b \\ y \end{bmatrix} = \begin{bmatrix} a & k_1' & \bar{b} \\ 0 & k_2' & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ w_b \\ u \end{bmatrix}, \tag{6.14}$$

where

$$k_1' = \frac{2\left(b^+ - \bar{b}\right)\left(\bar{b} - b^-\right)}{b^+ - b^-}$$

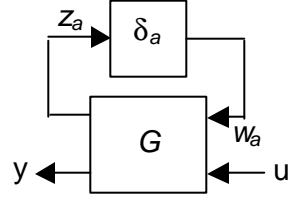$$k_2' = \frac{\left(b^+ + b^-\right) - 2\bar{b}}{b^+ - b^-}$$

Figure 6.1 Linear Fractional Transformation block diagram

Finally, combining the expressions for uncertainty in a and in b, the following state-space expression results. The accompanying block diagram (Figure 6.2) is a companion to Figure 6.1.

$$\begin{bmatrix} \dot{x} \\ z_a \\ z_b \\ y \end{bmatrix} = \begin{bmatrix} \bar{a} & k_1 & k_1' & \bar{b} \\ 1 & k_2 & 0 & 0 \\ 0 & 0 & k_2' & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ w_a \\ w_b \\ u \end{bmatrix}$$
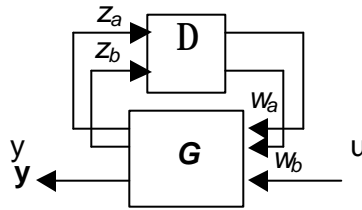
(6.15)

Figure 6.2 Companion to the Linear Fractional
Transformation block diagram

Now, Δ is no longer a scalar uncertainty but is a 2×2 diagonal matrix with normalized uncertainty terms.

$$\Delta = \begin{bmatrix} \delta_a & 0 \\ 0 & \delta_b \end{bmatrix} \tag{6.16}$$

Again, we have separated what is known from what is uncertain but bounded. Since Δ is no longer a scalar, we have to make another choice for a norm: the maximum singular value. It is not difficult to see that $\bar{s}(\Delta) \leq 1$. A very important observation is that the uncertain element Δ has a fixed structure: a diagonal matrix consisting of the individual uncertainties in a and b. Thus, unstructured uncertainty at the component level has become structured uncertainty at the system level. LFTs are the mathematical tools that allow us to provide this systemic structure for the uncertainty.

6.2.2   Other Types of Uncertainty Models

Other than the parametric uncertainty described in Section 6.2.1, there are at least two other types of uncertainty models. These are Unmodeled Dynamics or Uncertainty at the Input and Uncertainty at the Output models. The major difference between parametric uncertainty and the other two types of uncertainties is that parametric uncertainties are real-valued while the other types of uncertainties are complex-valued perturbations. The two types of uncertainty models are explained in the following subsections.

6.2.2.1   Unmodeled Dynamics (Uncertainty at the Input)

During the linearization process, higher-order terms in aircraft equations of motion are ignored. Also, other uncertainties arise due to aeroelasticity, control surface variations, and vehicle flexibility. Usually, the plant model is a good system representation term at low- to mid-frequency inputs, but modeling uncertainties become larger with high-frequency inputs. Instead of attempting to include all modeling uncertainties, the modeling uncertainties are treated as additives to the plant inputs. Figure 6.3 shows a general block diagram representing this approach to accounting for uncertainty at the plant input.
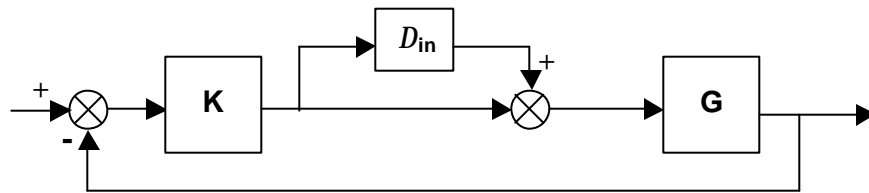

Figure 6.3 Unmodeled Dynamics block diagram

This type of uncertainty is parameterized with two elements, W_{in} and Δ_{in}. W_{in} is a weighting transfer function (assumed to be known) that reflects the amount of uncertainty in a model with respect to frequency. The other parameter, Δ_{in}, is a stable unknown transfer function that nevertheless satisfies the condition $||\Delta_{in}||_\infty < 1$.

6.2.2.2   Uncertainty at the Output

Similar to Unmodeled Uncertainty, uncertainty of the measurements is modeled as Uncertainty at the Output. A block diagram representing how this type of uncertainty is modeled is shown in Figure 6.4.
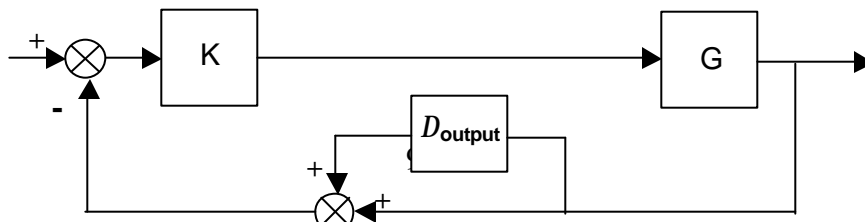
Figure 6.4 Uncertainty at the Output block diagram

6.2.3   *Structured Singular-Value Analysis (m-Analysis)*

Now that we know how to represent uncertainties in the system using LFTs, we must turn our attention to analyzing the robustness of systems modeled in this fashion. We follow the most common practice today by basing our analysis on the structured singular value, μ, and using available software tools to perform the μ-analysis. The technique is based on the following theorem:

$$Robust\ Stability \Leftrightarrow \mu_{\underline{\Delta}}\left(M_{11}(j\omega)\right) < 1 \quad \forall\ \omega \tag{6.17}$$

where M_{11} is the left upper corner block of M; i.e.,

$$M(j\omega) = \begin{bmatrix} M_{11}(j\omega) & M_{12}(j\omega) \\ M_{21}(j\omega) & M_{22}(j\omega) \end{bmatrix} \tag{6.18}$$

and the function $\mu_{\underline{\Delta}}$ is defined as

$$\mu_{\underline{\Delta}}(M) \equiv \frac{1}{min\left\{\overline{\sigma}(\Delta):\ \Delta \in \underline{\Delta}, det\left(I - M\Delta\right) = 0\right\}} \tag{6.19}$$

where $\underline{\Delta} = \left\{diag\left(\Delta_1,\ \Delta_2,\ ...,\ \Delta_n\right)\right\}$.

According to this theorem, $\mu_{\underline{\Delta}}$ is a function of M that depends on the structure of $\underline{\Delta}$. $m_{\underline{\Delta}}$ is the reciprocal of the smallest Δ (where we use $\overline{\sigma}$ as the norm) we can find for the set $\underline{\Delta}$ that makes the matrix $I - M\Delta$ singular. If no such Δ exists, $\mu_{\underline{\Delta}}$ is taken to be zero.

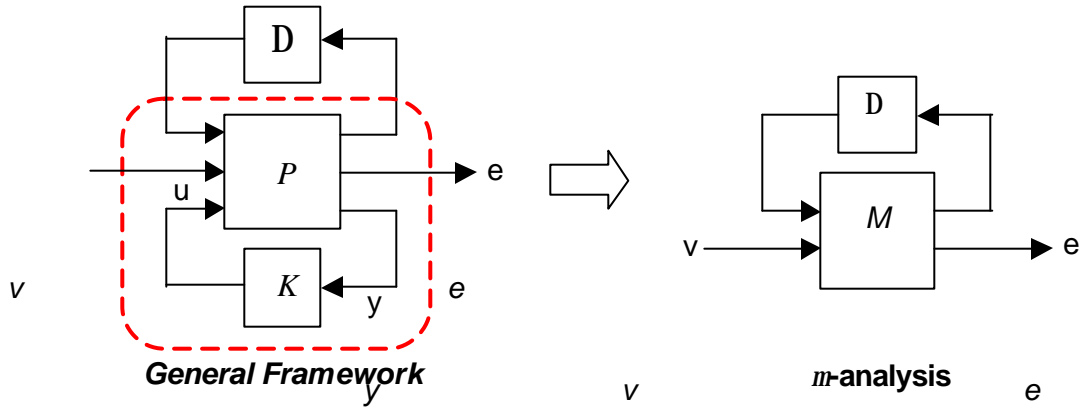The general framework and the μ-analysis transformation are shown in Figure 6.5.

Figure 6.5 General framework and μ-analysis transformation

Even though the function $\mu_{\underline{\Delta}}$ is defined, we must still calculate it. Unfortunately, no exact calculation algorithms exist. So, we must calculate its upper and lower bounds. Normally, the upper bound is used since these values of μ are "safer" (that is, they are more conservative). The upper bound is defined as

$$\mu_{\underline{\Delta}}(\boldsymbol{M}) \leq \inf_{\boldsymbol{D} \in \underline{\boldsymbol{D}}} \sigma\left(\boldsymbol{DMD}^{-1}\right), \tag{6.20}$$

where $\boldsymbol{D}$ is the scaling matrix. Figure 6.6 geometrically illustrates the effect of $\boldsymbol{D}$-scales.
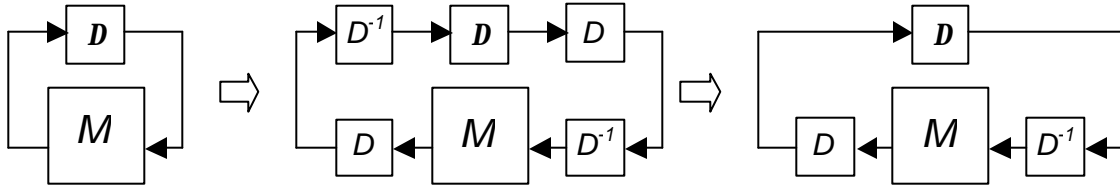

Figure 6.6 The effect of $\boldsymbol{D}$-scales

Another important feature of the upper bound is that it can be combined with the H∞ controller synthesis technique to yield a μ-synthesis method. Note that the upper bound, when applied to transfer functions and maximized across frequencies, is simply a scaled H∞ norm.

The steps needed to test the robust stability using μ-analysis are as follows:

1. Construct the interconnection structure, *M*, which is a known linear system.
2. Define a structured perturbation set, Δ.
3. Combine *M* and Δ to form the feedback system shown in Figure 6.5.
4. Calculate a frequency response of *M*.
5. Calculate the upper and lower bounds for μ.
6. Find the upper bound peak value.

95

7. If $\mu_{peak} < 1$: pass; if $\mu_{peak} > 1$: fail.

7   Bibliography

[1] Krener, A. J., "On the Equivalence of Control Systems and Linearizations of Nonlinear Systems," *SIAM Journal of Control and Optimization*, Vol. 11, 1973, pp. 670-676.

[2] Brockett, R. W., "Feedback Invariants for Nonlinear Systems," *Proceedings of VII IFAC Congress*, Helsinki, Finland, 1978, pp. 1115-1120.

[3] Falb, P. L. and Wolovich, W. A., "Decoupling in the Design and Synthesis of Multivariable Control Systems," *IEEE Transactions of Automatic Control*, AC-12, Vol. 6, 1967, pp. 651-659.

[4] Singh, S. N. and Rugh, W. J., "Decoupling in a Class of Nonlinear Systems by State Variable Feedback," *Transactions of ASME Journal of Dynamic Systems Measurements and Control*, Vol. 94, 1972, pp. 323-329.

[5] Freund, E., "Design of Time-Variable Multivariable Systems by Decoupling and by the Inverse," *IEEE Transactions of Automatic Control*, AC-16, Vol. 2, 1971, pp. 183-185.

[6] Porter, W. A., "Diagonalisation and Inverses for Nonlinear Systems," *International Journal of Control*, Vol. 10, 1970, pp. 252-264.

[7] Isidori, A., Krener, A. J., Giorgi, C., and Monaco, S., "Nonlinear Decoupling Via Feedback: a Differential Geometry Approach," *IEEE Transactions of Automatic Control*, AC-26, 1981, pp. 331-345.

[8] Byrnes, W. A., "Diagonalisation and Inverses for Nonlinear Systems," *International Journal of Control*, Vol. 10, 1970, pp. 252-264.

[9] Bugaski, D. J., Enns, D. F., and Elgersma, M. R., "A Dynamic Inversion Based Control with Application to the F-18 HARV," AIAA Paper 90-3407, *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Portland, OR, 1990, pp. 826-839.

[10] Honeywell Technology Center and Houston Engineering Center, "*Application of MACH to X-38 Drop Test Vehicle*," HTC Contract Number 7028327, for NASA Johnson Space Center, June 1997.

[11] Balas, G. J., Doyle, J. C., Glover, K., Packard, A., and Smith, R., μ-*Analysis and Synthesis TOOLBOX for Use with MATLAB*, The Mathworks Inc., First Edition, July 1993.

[12] Honeywell Technology Center, Lockheed Martin Skunk Works, and Lockheed Martin Tactical Aircraft Systems, "*Application of Multivariable Control Theory to Aircraft Control Laws: Final Report: Multivariable Control Design Guidelines*," WL-TR-96-3099, Flight Dynamics Directorate, Wright Laboratory, Air Force Materiel Command, Wright-Patterson AFB, OH 45433-7562, May, 1996.

[13] Balas, G.J., Garrard, W.L. and Reiner, J., "Robust Dynamic Inversion Control Laws for Aircraft Control," AIAA Paper 92-4329, *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Hilton Head Island, SC, Aug. 1992, pp. 192-205.

[14] Anon., "*Military Standard Flying Qualities of Piloted Aircraft,*" Mil-STD-1797A, Jan. 30, 1990.

[15] Enns, D., Bugajski, D., Hendrick, R. and Stein, G., "Dynamic Inversion: An Evolving Methodology for Flight Control Design," *International Journal of Control*, Vol. 59, No. 1, 1994, pp. 71-91.

[16] NASA Johnson Space Center, "Shuttle Engineering Simulator," July 1993.

[17] Snell, S.A. and Stout, P.W., "Robust Control of Angle of Attack Using Dynamic Inversion Combined with Quantitative Feedback Theory," AIAA Paper 96-3783, *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Hilton Head Island, SC, July 1996, pp. 1-12.

[18] Brinker, J.S. and Wise, K.A., "Stability and Flying Qualities Robustness of a Dynamic Inversion Aircraft Control Law," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 6, Nov.-Dec., 1996, pp. 1270-1277.

[19] Suikat, R., Donaldson, K. and Downing, D., "Analysis of a Candidate Control Algorithm for a Ride-Quality Augmentation System," *Journal of Guidance, Control, and Dynamics*, Vol. 12, No. 4, July-Aug. 1989, pp. 505-513.

[20] Ito, D. and Valasek, J., "Robust Dynamic Inversion Controller Design and Analysis to the X-38 Vehicle," AIAA Paper 2001-4380, *AIAA Guidance, Navigation and Control Conference*, Montreal, Canada, Aug. 2001.

[21] Doyle, J. C. and Stein, G., "Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis," *Proceedings of IEEE Transactions on Automatic Control*, Vol. AC-26, No. 1, February 1981, pp. 4-16.

[22] Lin, J. M., Chang, C. H. and Tsai, H. L., "Optimal Feedforward and Feedback Controllers Designed by LQG/LTR Method," *Journal of the Chinese Institute of Engineers*, Vol. 20, No. 1, 1997, pp. 57-66.

[23] Ridgely, D. B., Banda, S. S., McQuade, T. E. and Lynch, P. J. "Linear-Quadratic-Gaussian with Loop-Transfer-Recovery Methodology for an Unmanned Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 10, Jan.-Feb. 1987, pp. 82-89.

[24] Maciejowski, J. M., *Multivariable Feedback Design,* Addison-Wesley Publishing Company, 1992.

[25] Athans, M., "Tutorial on the LQG/LTR Method," AIAA Paper 97-3607, 1997, pp. 755-769.

[26] Anon., "*X-38 V131R FCS Requirements,*" NASA Johnson Space Center Internal Memo, May 2000.

[27] Stevens, B. L. and Lewis, F. L., *Aircraft Control and Simulation*, John Wiley & Sons, Inc., New York, 1992.

[28] Dang Vu, B., "Nonlinear Dynamic Inversion Control," *Lecture Notes in Control and Information Sciences 224: Robust Flight Control: A Design Challenge*, Springer, London, 1997, p. 111.

[29] Ghaoui, L. E., Carrier, A. and Bryson, A. E. "Linear Quadratic Minimax Controllers," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, July-Aug. 1992, pp. 953-961.

[30] Hess, R. A. and Snell, S. A. "Flight Control System Design with Rate Saturating Actuators," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, Jan.-Feb. 1997, pp. 90-96.

[31] Snell, S. A. and Hess, R. A. "Robust, Decoupled, Flight Control Design with Rate-Saturating Actuators," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 3, May-June 1998, pp. 361-367.

[32] Snell, S. A. "Decoupling Control Design with Applications to Flight," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 4, July-Aug. 1998, pp. 647-655.

[33] Tarbouriech, S., Garcia, G., and Henrion, D., "Local Stabilization of Linear Systems with Position and Rate Bounded Actuators," *Proceedings of 14th Triennial World Congress,* Beijing, P.R. China, F-2e-02-5, 1999, pp. 459-464.

[34] Nayfeh, A. H. and Balachandran, B., *Applied Nonlinear Dynamics*, John Wiley & Sons, Inc., New York, 1995, pp. 2-4.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE March 2002	3. REPORT TYPE AND DATES COVERED NASA Technical Paper	
4. TITLE AND SUBTITLE Reentry Flight Vehicle Controls Design Guidelines: Dynamic Inversion			**5. FUNDING NUMBERS**
6. AUTHOR(S) Daigoro Ito,* Jennifer Georgie,** John Valasek,** Donald T. Ward*			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Lyndon B. Johnson Space Center Houston, Texas 77058			**8. PERFORMING ORGANIZATION REPORT NUMBERS** S-881
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER** TP-2002-210771

11. SUPPLEMENTARY NOTES
*Lyndon B. Johnson Space Center, Houston, Texas
**Flight Simulation Laboratory, Texas Engineering Experiment Station, Texas A&M University, College Station, Texas

12a. DISTRIBUTION/AVAILABILITY STATEMENT Available from the NASA Center for AeroSpace Information (CASI) 7121 Standard Hanover, MD 21076-1320 Subject Category: 39	12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report addresses issues in developing a flight control design for vehicles operating across a broad flight regime and with highly nonlinear physical descriptions of motion. Specifically it addresses the need for reentry vehicles that could operate through reentry from space to controlled touchdown on Earth. The latter part of controlled descent is achieved by parachute or paraglider – or by an automatic or a human-controlled landing similar to that of the Orbiter. Since this report addresses the specific needs of human-carrying (not necessarily piloted) reentry vehicles, it deals with highly nonlinear equations of motion, and their generated control systems must be robust across a very wide range of physics. Thus, this report deals almost exclusively with some form of dynamic inversion (DI). Two vital aspects of control theory – noninteracting control laws and the transformation of nonlinear systems into equivalent linear systems – are embodied in DI. Though there is no doubt that the mathematical tools and underlying theory are widely available, there are open issues as to the practicality of using DI as the only or primary design approach for reentry vehicles. This report provides a set of guidelines that can be used to determine the practical usefulness of the technique.

14. SUBJECT TERMS flight control; spacecraft control; reentry vehicles; dynamics; equations of motion; nonlinear equations; control systems design		15. NUMBER OF PAGES 116	16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unlimited	20. LIMITATION OF ABSTRACT Unlimited